

4.4 Security and Accountability

This section describes the security and accountability tools used by DAAC operators:

1. DCE Cell Manager
2. TCP Wrapper
3. Crack
4. SATAN
5. TripWire
6. Tivoli (Enterprise Console and Admin)
7. Cryptographic Management Interface (CMI)

This page intentionally left blank.

4.4.1 DCE Cell Manager

The DCE Cell Manager (also referred to as “Chisholm DCE Cell Manager 1.6.2”) toolset can administer various components and interactions of a DCE cell including:

- **Namespace Manager** - manages a DCE cell’s directory namespace
- **Configuration Manager** - manages a DCE cell’s Distributed Time Service (DTS) processes and configuration
- **Security Manager** - manages a DCE cell’s security registry

DCE Cell Manager is used to perform the operator functions listed in Table 4.4.1-1.

Table 4.4.1-1. Common ECS Operator Functions Performed with DCE Cell Manager

Operating Function	GUI	Description	When and Why to Use
Organizing Cell Directory Service (CDS) Namespace Directories	Namespace Manager	Manages a DCE cell’s directory namespace.	When the need arises to manage, maintain, and monitor the cell namespace, keeping entries and access to them up to date.
Configuring the Basic Cell.	Configuration Manager	Manages a DCE cell’s configuration.	When a DCE cell’s configuration needs managing.
Synchronizing system clocks.	Configuration Manager	Manages a DCE cell’s Distributed Service tasks.	When system clocks need synchronizing.
Populating the Security Registry.	Security Manager	Manages a DCE cell’s security registry.	When the need arises, to: <ul style="list-style-type: none"> • allow addition, modification, and deletion of principal accounts for new users • create groups of principals to control access to registry and CDS objects • set policies on passwords and accounts • maintain replicas of the database that contains the registry

4.4.1.1 Quick Start Using DCE Cell Manager

To execute the DCE Cell Manager from the command line prompt type:

/opt/cellmgr/bin/cellManager

For more information on using DCE Cell Manager, see Section Starting and Using DCE Cell Manager, page 32 of the *Cell Manager 1.6. Overview and User’s Guide*.

4.4.1.2 DCE Cell Manager Main Screen GUIs

In order to use the DCE Cell Manager, the operator must first login to a DCE Cell (See Figure 4.4.1-1). For more information on starting the DCE Cell Manager , see section on Starting and Using DCE Cell Manager and Logging In, pages 32 through 34 of the *Cell Manager 1.6 Overview and User's Guide*.

STOP

CHISHOLM

Copyright (c) 1997 Chisholm Technologies Inc. All rights reserved.

DCE Login for cell: testcell.hitc.com

Login Name:

Login Password:

Log In Use Existing Ticket Cancel Help

Figure 4.4.1-1. DCE Login Dialog

4.4.1.2.1 DCE Cell Manager Launcher

If the login is accepted, the DCE Cell Manager Launcher will appear as shown in Figure 4.4.1-2. If the login name or password is not accepted, a window will pop up with an appropriate error message and the operator may try again.

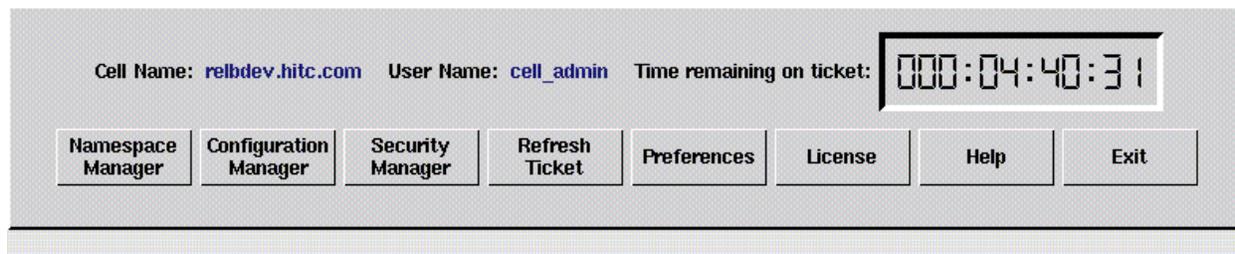


Figure 4.4.1-2. DCE Cell Manager Launcher

The DCE Cell Manager provides the following options:

- The **Namespace Manager** button brings up the Namespace Manager GUI. The initial display of this window is a tree structure with the cell root entries and a first level of directory entries. For more information, Cell Manager 1.6 Namespace Manager User's Guide.
- The **Configuration Manager** button brings up the Configuration Manager GUI which lists the hosts currently configured in the cell. For more information, Cell Manager 1.6 Configuration Managers User's Guide.
- The **Security Manager** button brings up the Security Manager GUI showing the organizations in the registry and represents the organization domain. For more information, Cell Manager 1.6 Security Manager User's Guide.
- The **Preferences** button brings up the DCE Launcher Preferences GUI. From this GUI additional buttons can be added to the Launcher window for applications commonly used in a DCE management session. Buttons on the DCE Cell Manager Launcher window can also be replaced or deleted. For more information, pages 36 through 39 of the Cell Manager 1.6 Overview and User's Guide.
- The **Refresh Ticket** button brings up a GUI where the DCE password must be entered. It is use to refresh the time remaining on the Authentication Ticket used by the DCE Cell Manager and which is indicated on the upper right box. For more information, see pages 35-36 of the Cell Manager 1.6 Overview and User's Guide.

- The **License** button brings up the Cell Manager License GUI. It is used to view, edit or load the License Agreement purchased. Not, the License can only be viewed or modified by the Super User or Systems Administrator. For more information, see pages 39 through 42 of the Cell Manager 1.6 Overview and User's Guide.
- The **Help** buttons brings up a display with concise information about specific windows, fields, buttons, and other application components.
- The **Exit** button provides options for exiting, including closing the DCE Cell Manager Launcher window.

4.4.1.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for DCE Cell Manager, refer to the ECS Baseline Information System web page, URL

<http://cmdm.east.hitc.com/>.

The document is 914-TDA-058, "DCE Cell Manager Installation and Configuration Report"

4.4.1.4 Databases

None.

4.4.1.5 Special Constraints

None.

4.4.1.6 Outputs

The DCE Cell Manager provides the option of recording output from any DCE commands and commands executed during a current session of the tool and saving them to a history file. For more information, see Appendix B: Common Features of DCE Cell Manager Tools, *Cell Manager 1.6 Overview and User's Guide*.

4.4.1.7 Event and Error Messages

None.

4.4.1.8 Reports

None.

4.4.2 TCP Wrappers

TCP Wrappers allow the operator to control access to various network services through the use of access control lists. They also provide logging information of wrapped network services which may be used to prevent or monitor network attacks. It intercepts incoming network connections and verifies if the connection is allowed before passing the connection onto the actual network daemon. TCP Wrappers allows the operator to monitor and filter incoming requests for the systat, finger, ftp, telnet, rlogin, rsh, exec, tftp, talk, and other network services. Full descriptions of these Unix services can be obtained using the “man” command, e.g., man systat. TCP Wrappers performs the following functions automatically:

- **Access control:** access can be controlled per host, per service, or combinations thereof.
- **Host name spoofing:** verifies the client host name that is returned by the address->name DNS server, by asking for a second opinion from a local DNS server.
- **Host address spoofing:** the wrapper programs can give additional protection against hosts that claim to have an address that lies outside their own network.
- **Client username lookups:** the protocol proposed in RFC 931 provides a means to obtain the client user name from the client host. The requirement is that the client host runs an RFC 931-compliant daemon. The information provided by such a daemon is not intended to be used for authentication purposes, but it can provide additional information about the owner of a TCP connection.
- **Multiple ftp/gopher/www archives on one host:** `daemon@host' access control patterns can be used to distinguish requests by the network address that they are aimed at. Judicious use of the `twist' option (see the hosts_options.5 file supplied with TCP Wrappers, `nroff -man' format) can guide the requests to the right server. These can be servers that live in separate chroot areas, or servers modified to take additional context from the command line, or a combination.
- **Sequence number guessing:** client username lookup protocol can help to detect host impersonation attacks. Before accepting a client request, the wrappers can query the client's IDENT server and find out that the client never sent that request.

Additional information on TCP Wrappers can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-firewall.html>

TCP Wrappers is used to perform the operator functions listed in Table 4.4.2-1.

Table 4.4.2-1. Common ECS Operator Functions Performed with TCP Wrappers

Operating Function	Command/Action	Description	When and Why to Use
Monitor potentially malicious attempts to access network services.	Check TCP Wrappers log using a text editor.	Program continuously runs in the background appearing to malicious external client service requests as a normal inetd daemon process.	To check for evidence of an attempt of breaking-in.

4.4.2.1 Quick Start Using TCP Wrappers

TCP Wrappers provides a library of tiny daemon wrapper programs. The daemons each correspond to a service provided by the host operating system. The daemons are registered with the service, which results in the operating system invoking the daemon each time that service is invoked. The daemons perform their function(s) and terminate. A common function is to log the name of the client host and requested service. They do not exchange information with client or server applications, and impose no overhead on the actual conversation between the client and server applications. Optional features include: access control to restrict what systems can connect to what network daemons; client user name lookups with the RFC 931 protocol; additional protection against hosts that pretend to have someone else's host name; and additional protection against hosts that pretend to have someone else's host address.

4.4.2.1.1 Command Line Interface

The TCP Wrappers cannot be invoked or accessed from the command line. The TCP Wrapper daemons are invoked by the operating system service to which they are registered. The daemons terminate upon completing their function.

4.4.2.2 TCP Wrapper Main Screen

TCP Wrapper does not have a graphical user interface.

4.4.2.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for AMASS, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

4.4.2.4 Databases

None.

4.4.2.5 Special Constraints

None.

4.4.2.6 Outputs

Table 4.4.2-2 describes TCP Wrappers output.

Table 4.4.2-2. Outputs

Output	Disposition	Description and Format
Event log	The disposition of the wrapper logs is determined by the system configuration file parameter for the system log file name. In the ECS Release 4 Development Environment (host = mss1) the parameter is <i>local3.info</i> and the log file is <i>/etc/syslog.conf</i> .	The Wrapper daemons log the event/service request that caused their invocation. The log provides sufficient information to describe the event/service request and response from the system. Log records are output in ASCII text format, each record containing the following fields: <ul style="list-style-type: none">• Date• Time• host• Service• Response to request• Event

Figure 4.4.2.1 shows an example of a log file created by TCP Wrappers. The log file may be examined with available tools like the **vi** editor. The contents of the log file may be used to generate reports on Service Request activity for the host.

```
Nov 5 14:49:37 tle2sun in.telnetd[584]: connect from tle2sun.HITC.COM
Nov 5 19:59:26 tle2sun in.rshd[5283]: connect from neptune.HITC.COM
Nov 5 19:59:29 tle2sun in.rshd[5318]: connect from neptune.HITC.COM
Nov 6 10:46:20 tle2sun in.telnetd[6417]: connect from ins1.HITC.COM
Nov 6 11:18:38 tle2sun in.rlogind[6608]: connect from ins1.HITC.COM
Nov 6 13:03:24 tle2sun in.rlogind[6739]: connect from mss1.HITC.COM
Nov 6 13:06:33 tle2sun in.rlogind[6758]: connect from neptune.HITC.COM
Nov 6 13:06:40 tle2sun in.rshd[6761]: connect from neptune.HITC.COM
Nov 6 13:07:03 tle2sun in.rshd[6763]: connect from neptune.HITC.COM
Nov 6 13:07:22 tle2sun in.rshd[6765]: connect from neptune.HITC.COM
Nov 6 14:37:42 tle2sun in.rshd[6860]: connect from stargazer.HITC.COM
Nov 6 16:19:17 tle2sun-e0 in.rlogind[381]: connect from ins1.HITC.COM
Nov 6 19:39:50 tle2sun-e0 in.rlogind[674]: connect from ins1.HITC.COM
Feb 14 16:21:09 vulcan in.rlogind[163]: connect from akashi.HITC.COM
Feb 14 17:17:01 vulcan in.rlogind[367]: connect from akashi.HITC.COM
Feb 14 17:28:35 vulcan in.rshd[371]: connect from huckfinn.HITC.COM
Feb 14 17:40:48 vulcan in.rlogind[372]: connect from dss1.HITC.COM
Feb 14 17:47:28 vulcan in.telnetd[379]: connect from dss1.HITC.COM
Feb 14 18:43:07 vulcan in.telnetd[395]: refused connect from spg-as14s67.erols.com
Feb 14 19:29:00 vulcan in.rlogind[405]: connect from dss1.HITC.COM
Feb 16 05:52:42 vulcan in.rlogind[705]: connect from boston.HITC.COM
Feb 16 19:45:33 vulcan in.telnetd[676]: refused connect from spg-as27s43.erols.com
Feb 17 07:40:49 vulcan in.rshd[1040]: connect from neptune.HITC.COM
Feb 17 19:33:36 vulcan in.rshd[595]: connect from transam.gsfc.nasa.gov
Feb 18 09:34:11 vulcan in.telnetd[858]: connect from hobbes.HITC.COM
Feb 18 10:11:38 vulcan in.telnetd[958]: connect from hobbes.HITC.COM
Feb 18 12:39:20 vulcan in.rlogind[1025]: connect from klingon.HITC.COM
Feb 18 12:40:37 vulcan in.rshd[1046]: connect from neptune.HITC.COM
Feb 18 12:40:44 vulcan in.rshd[1049]: connect from neptune.HITC.COM
Feb 18 12:40:50 vulcan in.rshd[1051]: connect from neptune.HITC.COM
Feb 18 12:40:55 vulcan in.rshd[1054]: connect from neptune.HITC.COM
```

Figure 4.4.2-1. Example of TCP Wrappers Log

The log file provides the following information for each entry: data and time; host sever name; type of service requested and port that provides that service; answer given to the request connection (connect/refused); client host name.

4.4.2.7 Event and Error Messages

None.

4.4.2.8 Reports

None.

4.4.3 Crack

Crack is a COTS freeware product used in compliance management. Crack is designed to find standard Unix eight-character Data Encryption Standard (DES) encrypted passwords by standard guessing techniques outlined below. It is flexible, configurable and fast, and able to make use of several networked hosts via the Berkeley rsh program (or similar), where possible.

Crack takes as input a series of password files and source dictionaries. It merges the dictionaries, turns the password files into a sorted list, and generates lists of possible passwords from the merged dictionary or from information gleaned about users from the password file. It does not attempt to remedy the problem of allowing users to have guessable passwords, and it should not be used in place of getting a really good, secure passwd program replacement.

Crack works by making many individual passes over the password entries that are supplied to it. Each pass generates password guesses based upon a sequence of rules, supplied to the program by the user. The rules are specified in a simplistic language in the files `gecos.rules` and `dicts.rules`, to be found in the Scripts directory. The rules are written as a simple string of characters, with one rule to a line. Blank lines, and comment lines beginning with a hash character # are ignored. Trailing whitespace is also ignored. The instructions in the rule are followed from left to right, and are applied to the dictionary words one by one, as the words are loaded. Some simple pattern matching primitives are provided for selection purposes, so that if the dictionary word does not match the pattern, it is ignored.

Additional information on Crack can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-firewall.html>

Crack is used to perform the operator functions listed in Table 4.4.3-1.

Table 4.4.3-1. Common ECS Operator Functions Performed with Crack

Operating Function	Command	Description	When and Why to Use
Ensure passwords used aren't easy to guess.	<code>crack filename</code>	Scans the password file and makes guesses of passwords.	Regularly review systems for weak passwords.

4.4.3.1 Quick Start Using Crack

For more information on Crack, see the `readme.txt` file in the Docs directory where Crack is installed. The directory where Crack is installed may be found in the ReadMe file described in Section 4.4.3.3 Required Operating Environment.

The documentation of Crack used as a basis and referenced in this section is for version 4.1.

4.4.3.1.1 Invoking Crack From the Command Line Interface

To execute Crack from the command line prompt use:

```
Crack [filename]
```

Where *[filename]* is the name of the password file.

4.4.3.2 Crack Main Screen

Crack does not have a GUI. Crack output is returned to the command line interface where Crack was started. The Crack startup message and initialization is shown in Figure 4.4.3-1 below.

```
mss2
mss2{rsnyder}>71: ypcat passwd > ypfile
mss2{rsnyder}>72: ./Crack -nice 10 ypfile
Crack 4.1f RELEASE, The Password Cracker (c) Alec D.E. Muffett, 1992
Invoked as: ./Crack -nice 10 ypfile
Making dictionary Dicts/bigdict - This may take some time...
touch Dicts/.lockfile
Binary directory: /home/rsnyder/COTS/security/crack_4.1/10
( cd ../Sources ; make clean )
make[1]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
rm -f *.o *.u *.a *.pixie *.Addrs *.Counts
rm -f crack-pwc tester bytesex testrule
rm -f speedcrypt speedfcrypt speedxform speedufc
make[1]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
( cd ../Sources ; make crack-pwc,which )
make[1]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
Choosing between Crack,fcrypt and Crack,ufc
((../Scripts/do_ufc && make crack-pwc,ufc) || make crack-pwc,fcrypt)
Looking for UFC-crypt in /home/rsnyder/COTS/security/crack_4.1/ufc-crypt
Cannot find /home/rsnyder/COTS/security/crack_4.1/ufc-crypt - cannot use UFC-cry
pt on this platform
make[2]: Entering directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
cc -O -c crack-lib.c -o crack-lib.o
cc -O -c crack-glob.c -o crack-glob.o
cc -O -c crack-supp.c -o crack-supp.o
cc -O -c crack-sort.c -o crack-sort.o
cc -O -c bytesex.c -o bytesex.o
"bytesex.c", line 19: warning: shift count negative or too big: <= 32
cc -O -o bytesex bytesex.o
cc -O `./bytesex` -c crack-fcrypt.c
cc -O -c crack-pwc.c -o crack-pwc.o
"crack-pwc.c", line 1103: warning: argument #2 is incompatible with prototype:
  prototype: pointer to function(int) returning void : "/usr/include/signa
l.h", line 37
  argument : pointer to function(void) returning void
cc -O -o crack-pwc crack-pwc.o crack-lib.o crack-glob.o crack-supp.o crack-sort.
o crack-fcrypt.o
make[2]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
make[1]: Leaving directory `/home/rsnyder/COTS/security/crack_4.1/Sources'
cp ../Sources/crack-pwc .
Sorting data for Crack.
Flags: -nice -i /tmp/pw.24133 Dicts/bigdict
Running program in background
Output will be written to a file in directory /home/rsnyder/COTS/security/crack_
4.1
named 'out<something>'
mss2{rsnyder}>73: █
```

Figure 4.4.3-1. Xterm window of Crack startup message and initialization

4.4.3.3 Required Operating Environment

Crack can run on any ECS platform. However, due to the possibility of security breaches and the high overhead in using the tool, it is implemented on only a few select machines.

Additional information regarding Crack is stored in a CM controlled ReadMe file. To find the ReadMe file for Crack, use the XRP Baseline Manager to determine where in ClearCase the ReadMe file resides

4.4.3.4 Databases

The source dictionaries used by Crack are gecos.rules and dicts.rules. These files will be maintained at the SMC and updates periodically forwarded to the DAACs. Instructions for updating these files are included in the Crack Readme file that is located in the Crack directory.

4.4.3.5 Special Constraints

None.

4.4.3.6 Outputs

Table 4.4.3-2 describes the output for Crack.

Table 4.4.3-2. Outputs

Output	Description and Format
Report of a person's user id and password used to get into the system	The results of running Crack are written to the file specified when Crack is invoked and can be shown on the screen in ASCII format

4.4.3.7 Event and Error Messages

None.

4.4.3.8 Reports

Crack 4.1 does not support reports.

4.4.4 SATAN

The Security Administrator Tool for Analyzing Networks (SATAN) is a COTS freeware product used for security compliance management. The tool is used to evaluate DAAC components for common security deficiencies and provide recommendations to enhance security.

SATAN gathers information about specified hosts and networks by examining network services (for example, finger, NFS, NIS, ftp, and rexd). It can then report this data in a summary format or, with a simple rule-based system, investigate potential security problems. Problems are described briefly and pointers provided to patches or workarounds. In addition to reporting vulnerabilities, SATAN gathers general network information (network topology, network services run, types of hardware and software being used on the network). SATAN has an exploratory mode that allows it to probe hosts that have not been explicitly specified. Thus, SATAN can probe not only targeted hosts, but also hosts outside your administrative domain.

SATAN is used to perform the operator functions listed in Table 4.4.4-1.

Table 4.4.4-1. Common ECS Operator Functions Performed with SATAN

Operating Function	Command	Description	When and Why to Use
Check systems for security leaks/holes	<code>satan machinename</code>	Checks NFS export to unprivileged programs. Checks NFS export via portmapper. Checks Unrestricted NFS export. Checks NIS password file access. Checks rexd access. Checks Sendmail vulnerabilities. Checks TFTP file access. Checks Remote shell access. Checks Unrestricted X server access. Checks Writable FTP home directory. Checks wu-ftpd vulnerability.	Whenever a security event has been detected or on a scheduled basis.

4.4.4.1 Quick Start Using SATAN

To execute SATAN from the command line prompt, enter:

```
>satan
```

The SATAN home page is shown in Figure 4.4.4-1 below.

The SATAN home page is also the GUI used by SATAN to communicate with the operator. When SATAN initializes, it starts its own local web server. Operators can then use Netscape (See Netscape Navigator, Section 4.12.5) to interact with the program. The URL for SATAN is provided in the ReadMe file (See Required Operating Environment, Section 4.4.4.3 below).

Additional information regarding SATAN is available from the link on the home page and the SATAN help files.

The documentation of SATAN used as a basis and referenced in this section is for version 1.1.1. Additional information on SATAN can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-network.html>

This command launches an Xterm window with the SATAN startup message shown in Figure 4.4.4-1 below.

The image shows a screenshot of an Xterm window. The window title bar is dark blue with the text 'mss2' in white. The window content is white with black text. The text displayed is: 'mss2{rsnyder}38: ./satan', 'SATAN is starting up....', '[1] 18367', and 'mss2{rsnyder}39:'. There is a cursor at the end of the last line.

```
mss2{rsnyder}38: ./satan
SATAN is starting up....
[1] 18367
mss2{rsnyder}39: █
```

Figure 4.4.4-1. Xterm window of SATAN showing startup message

After SATAN starts, the operator may use Netscape Navigator to communicate with the program through ECS SATAN home page (see Figure 4.4.4-2 below).

4.4.4.2 SATAN Main Screen

After the SATAN startup message shown in Figure 4.4.4-1 is displayed the SATAN home page (Figure 4.4.4-2) is available from the local web server. The operator may access on-line help for all SATAN functions using the SATAN Documentation link.

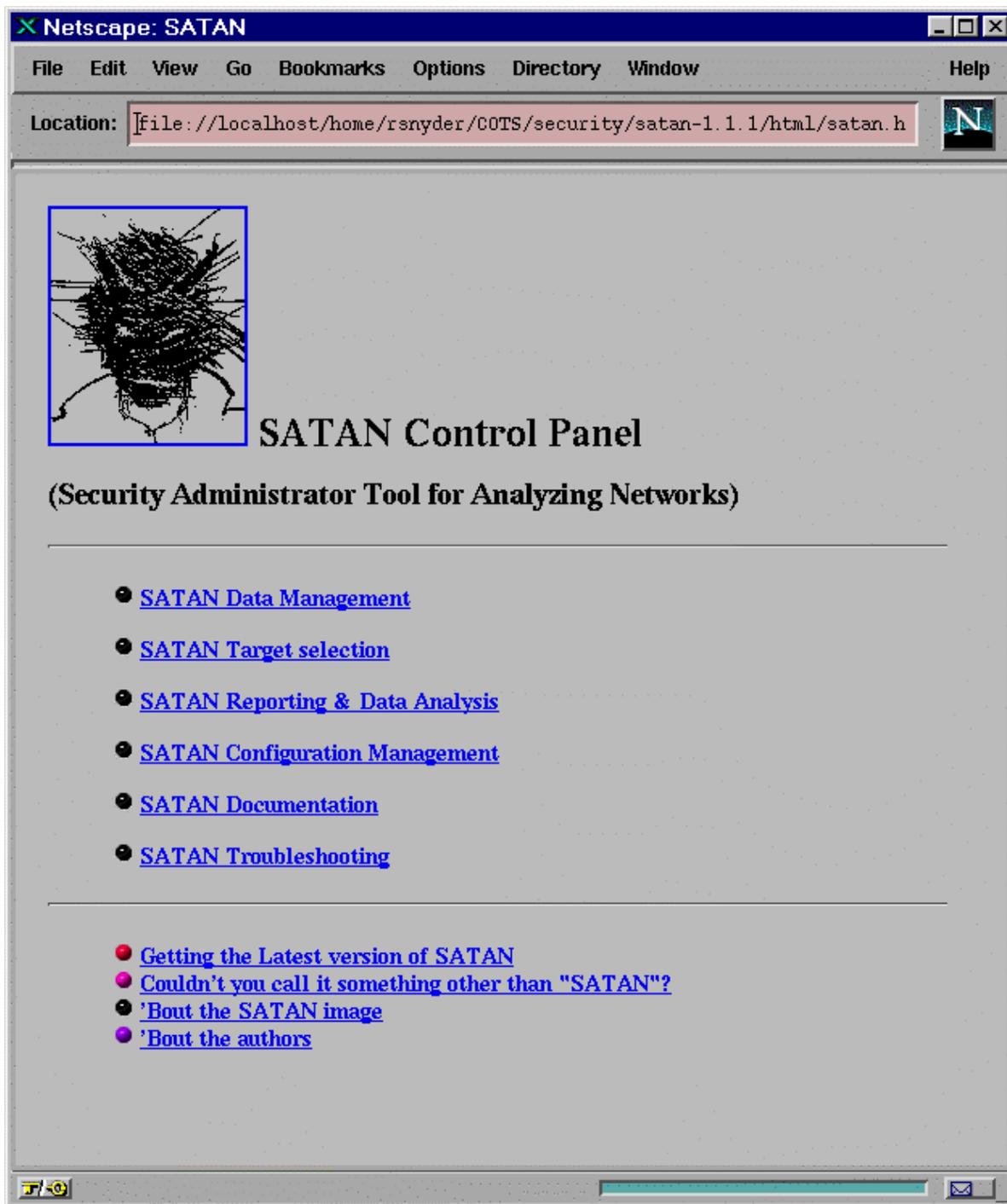


Figure 4.4.4-2. SATAN web home page

4.4.4.3 Required Operating Environment

SATAN requires UNIX. SATAN can run on the following platforms: HP J210, Sun Sparc 20/50, and Sun Sparc 20/71. However, due to the potential for security breaches with uncontrolled use, it is implemented on a few select machines.

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for SATAN, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

4.4.4.4 Databases

The table of common deficiencies used by SATAN is proprietary. This table provides the specific tests SATAN uses to determine “deficiencies”. Modifications to this table are discussed in the on-line documentation available through the SATAN home page (Figure 4.4.4-2).

4.4.4.5 Special Constraints

The amount of memory that SATAN uses can be large, depending on how many hosts are being scanned in or are in the database, so it should be run at off-peak hours.

4.4.4.6 Outputs

SATAN outputs are available using the “SATAN Reporting & Data Analysis” selection on the SATAN home page (Figure 4.4.4-2).

4.4.4.7 Event and Error Messages

None.

4.4.4.8 Reports

See Section 4.4.4.6 above.

4.4.5 Tripwire

Tripwire is an intrusion detection tool that aids system administrators and users in monitoring a designated set of files for any changes. File systems may be altered without authorization in a number of ways, including an intruder, an authorized user violating a DAAC policy, or malicious code-altering system executables as others are run. Using Tripwire, unauthorized changes are tracked in very little time.

Tripwire automates the creation of input lists and output lists of files. Tripwire uses the file `tw.config` to maintain the list of tested files. File attributes such as file size, ownership, inode number, inode values and timestamps are compared between the input and output lists. For each file, Tripwire computes a digital signature which is a fixed-sized output generated by a signature function whose input is an arbitrary file. If the contents of a file are changed in any way, then the signature will also change. One of the signature functions is to test for the integrity of a file system by generating checksums of files and comparing them with a previously generated database of checksums. Added or deleted files are flagged and reported, as are any files that have changed from their previously recorded state in the database. When run against system files on a regular basis, any file changes would be spotted when Tripwire is next run, giving system administrators information to enact damage control measures immediately.

Tripwire uses message-digest algorithms (one-way hash functions) to detect changes in a hard-to-spoof manner. This will be able to detect significant changes to critical files, including those caused by insertion of backdoor traps or viruses. Tripwire also monitors changes to file permissions, modification times, and other significant changes to inodes as selected by the system administrator on a per file/directory basis. Tripwire performs the following functions automatically:

- **Database Generation** -- Tripwire initializes the database based upon the entries enumerated in the `tw.config` file.
- **Database Update** -- Provides incremental database update functionality on a per-file/directory basis. This obviates having to regenerate the entire database every time a file or set of files change.
- **Integrity Checking** -- Generates a report of added, deleted, or changed files, comparing all the files described by the `tw.config` file against the files residing on the file system.
- **Interactive Update** -- Reports added, deleted, and changed files and prompts the user whether those database entries should be updated. The Interactive Update provides a method for system administrators to keep Tripwire databases "in sync" with file systems that change.

Tripwire is used to perform the operator functions listed in Table 4.4.5-1.

Table 4.4.5-1. Common ECS Operator Functions Performed with SATAN

Operating Function	Command	Description	When and Why to Use
Change the configuration file.	Edit the specific configuration file using the vi editor.	Specify which file(s) should be monitored.	<ul style="list-style-type: none"> • when another file needs to be monitored • checks the integrity of the file system specified when the daemon is started
Compare file signatures with database.	Done by Tripwire "chron" run periodically	Compares files current signatures against the data base and emails the operator a notification for changed files.	<ul style="list-style-type: none"> • This activity is a continuous, periodic performed on a configured interval by the "chron" run
Update the signatures datastore.	Done manually in response to "Interactive Update" prompts	Updates the signature datastore when the email notification discloses legitimate changes.	<ul style="list-style-type: none"> • As necessary to maintain a valid datastore of signatures.

4.4.5.1 Quick Start Using Tripwire

The following command is used to execute Tripwire from the command line prompt:

```
/etc/Tripwire-1.2/src/tripwire -v > {filename}
```

The following is the general syntax of executing Tripwire

```
tripwire [ options ... ] >filename
```

Where *'options'* are:

- initialize Database Generation mode -init
- update entry update entry (a file, directory, or tw.config entry) in the database
- interactive Integrity Checking mode with Interactive entry updating
- loosedir use looser checking rules for directories
- d dbasefile read in database from dbasefile (use ``-d -'` to read from stdin)
- c configfile read in config file from configfile (use ``-c -'` to read from stdin)
- cfd fd read in config file from specified fd
- dfd fd read in the database file from specified fd
- Dvar=value define a tw.config variable (ala @@define)
- Uvar undefine a tw.config variable (ala @@undef)

-i #|all ignore the specified signature (to reduce execution time)
-q quiet mode
-v verbose mode
-preprocess print out preprocessed configuration file
-E save as -preprocess
-help print out interpretation help message
-version print version and patch information

filename is a complete filename (including path) for the output report file.

Tripwire is automatically invoked on all machines by a “chron” run which periodically executes Tripwire. The operator receives information from Tripwire by email for files whose current signature does not match the datastore signature. The operator must verify the file changes and update the datastore or report a security violation. Tripwire may be run manually to update the datastore or create reports. The Operator can also generate Tripwire reports via the command line in the Xterm.

Additional information on Tripwire can be obtained at the following URL:

<http://www.alw.nih.gov/Security/prog-monitor.html>

The documentation of Tripwire, used as a basis and referenced in this section, is for version 1.2.

The differences between the behavior of Tripwire started from the “Chron” run and started by the operator result from the use of appropriate parameters on the start command. These parameters are listed and explained below.

4.4.5.2 Tripwire Main Screen

Tripwire does not have a GUI. The Tripwire startup message is shown in Figure 4.4.5-1 below.

```

X mss2
mss2{rsnyder}162: ./tripwire -i 1 -i 2
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
### Phase 3:  Creating file information database
### Phase 4:  Searching for inconsistencies
###
###
###          Total files scanned:          2867
###          Files added:                  0
###          Files deleted:                0
###          Files changed:                2677
###
###          After applying rules:
###          Changes discarded:            2675
###          Changes remaining:           2
###
changed: -rw----- rsnyder   499198 Apr 23 11:09:42 1998 /home/rsnyder/COTS/se
curity/tripwire-1.2/src/databases/tw.db_mss2
changed: -rwxr-xr-x rsnyder    4725 Apr 23 11:11:09 1998 /home/rsnyder/./cshrc
### Phase 5:  Generating observed/expected pairs for changed files
###
### Attr      Observed (what it is)      Expected (what it should be)
### =====
/home/rsnyder/COTS/security/tripwire-1.2/src/databases/tw.db_mss2
    st_size: 499198                      483328
    st_mtime: Thu Apr 23 11:09:42 1998    Thu Apr 23 11:09:36 1998
    st_ctime: Thu Apr 23 11:09:42 1998    Thu Apr 23 11:09:36 1998

/home/rsnyder/./cshrc
    st_mtime: Thu Apr 23 11:11:09 1998    Tue Apr 21 10:08:39 1998
    st_ctime: Thu Apr 23 11:11:09 1998    Tue Apr 21 10:08:39 1998

mss2{rsnyder}163: █

```

Figure 4.4.5-1. Xterm window with Tripwire showing Tripwire startup message.

4.4.5.3 Required Operating Environment

Tripwire runs on all Sun, SGI, and HP servers.

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for Tripwire, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

4.4.5.4 Databases

Tripwire uses a datastore of captured information. Tripwire must be started from the command line interface to manually update this datastore with the information emailed to the operator.

4.4.5.5 Special Constraints

None.

4.4.5.6 Outputs

Tripwire generates the outputs presented in Table 4.4.5-2 below in the filename specified on the command line invocation. A sample of the generated report is shown in section 4.4.5.8.

Table 4.4.5-2. Outputs

Output	Description and Format
Tripwire compares the new datastore with the existing Tripwire datastore stored on the file system, reporting added or deleted files, as well as those files that have changed.	See 4.4.5.8, the Report section, for a sample of Tripwire output.
Email to the operator.	Email messages list the files examined by Tripwire whose current signature does not match the file's entry in the signature file.
Updates to the Tripwire datastore.	The operator must review the email mentioned above and determine whether it represents a data corruption problem or that the Tripwire signature datastore is out-of-date. If the determination is that the datastore is out of date the operator must use Tripwire with the interactive update option and update the signature file.
Security problem notification	If the operator determines the Tripwire email indicates a security violation the operator must log the problem.

4.4.5.7 Event and Error Messages

None.

4.4.5.8 Reports

Tripwire must be started from the command line interface to request the Tripwire report. A sample of Tripwire output is shown below in Figure 4.4.5-2.

2:30am (mentor) 985 % Tripwire

```
### Phase 1: Reading configuration file
### Phase 2: Generating file list
### Phase 3: Creating file information database
### Phase 4: Searching for inconsistencies
###
###           Total files scanned:      82
###           Files added:              0
###           Files deleted:            0
###           Files changed:           80
###
###           After applying rules:
###           Changes discarded:       79
###           Changes remaining:       1
###
changed: -rw----- genek 4433 Oct 13 02:30:34 1992 /tmp/genek/Tripwire-0.92/config.h
### Phase 5: Generating observed/expected pairs for changed files
###
### Attr      Observed (what it is)      Expected (what it should be)
### =====
/tmp/genek/Tripwire-0.92/config.h
st size: 4441                4433
md5 (sig1): 0aqL1O06C3Fj1YBXz3.CPdcb    0cPX1H.DYS.s1vZdKD.ELMDR
snefru (sig2): 0PcgcK/MZvEm.8pIWe.Gbnn/  /8VoJv1JcoUA0NvoGN.k3P6E
crc32 (sig3): .EHA6x                /OuGNV
crc16 (sig4): ...9/q                ...6yu
md4 (sig5): /hQ0sU.UEbJo.UR4VZ/mNG/h    .UR4VZ/mNG/h/VSG/W/Z643k
md2 (sig6): .hLwjb.VRA0O.Z72y90xTYqA    1LR0Gg1l.vqB0.1g330P18/p
```

Figure 4.4.5-2. Tripwire Report

4.4.6 Tivoli Enterprise Console and Administration

Tivoli Enterprise Console (TEC) and Tivoli/Admin are used by DAAC Operators to perform security functions. For additional information about Tivoli, see Section 4.2.2, “Tivoli” and Section 4.3.5 Tivoli/Courier.

The TEC is used to collect and monitor security events. Through the TEC the operator is able to monitor a users successful/unsuccessful login attempt, a successful/unsuccessful DCE login attempt, file access denial, number of unsuccessful login attempts, etc. TEC refers to the collection of capabilities that can be performed on managed objects. There is no specific graphic or visualization of “The Tivoli Enterprise Console”. For more information, see the *Tivoli/ Enterprise Console Event Adapter Guide* and the *Tivoli/Enterprise Console User’s Guide*.

The Tivoli Desktop is used to provide access and permissions to various applications. Tivoli provides the ability to create a unique desktop for each user. Thus enabling each user to have a unique set of available applications.

The Tivoli/Admin is used for the management/creation of UNIX accounts, UNIX Groups, and host mappings. The Tivoli/Admin GUI (the Tivoli Main Screen, Figure 4.4.6-1) provides the ability to create a DCE account and map the user to their assigned roles. For more information, see the *Tivoli User and Group Management Guide* and the *Tivoli Host Management Guide*.

The most frequently used Enterprise Console and Tivoli/Admin functions are listed in Table 4.4.6-1.

Table 4.4.6-1. Common ECS Operator Functions Performed with Tivoli/Admin and TEC

Operating Function	Command/Script or GUI	Description	When and Why to Use
Create Event Console.	Series of GUIs.	Create an event console.	Use to allow the operator to view selected events that are received by the Event Server.
Create user account.	Series of GUIs.	Create a definition of a new user.	Use when there is a new user account to be created.
Manage NIS maps.	Series of GUIs.	Distribute software files.	Use to distribute software to specified hosts.
Manage Unix host.	Series of GUIs.	Allows centralized administration of a Tivoli managed host.	Use to check the status of a host or to modify certain parameters.
View Event.	GUI	View an event displayed in the Event Console.	To obtain more detailed information concerning an event in the Event Console.

4.4.6.1 Quick Start Using Tivoli/Admin

This section presents an orientation of Tivoli/Admin. The underlying assumptions are that:

- Tivoli and Tivoli/Admin has been installed on all of the platforms that is involved in the administration process
- The user has been granted administrator privileges for Tivoli/Admin activity.

The documentation of Tivoli used as a basis and referenced in this section is for version/release 3.0.1, contained in the ECS Release 4.

4.4.6.1.1 Invoking Tivoli/Admin and Tivoli Enterprise Console from the Command Line Interface

Both Tivoli/Admin and Tivoli Event Console are started from the main Tivoli GUI. The icons for each specific application is described in detail in later sections of this document.

To execute Tivoli from the command line prompt use:

```
source /etc/setup_env.csh (in c shell)
./etc/Tivoli/setup_env.sh (in Bourne or bash shell)
tivoli -font fixed
```

4.4.6.2 Tivoli Main Screen

The Main Screen as shown in Figure 4.4.6-1 facilitates software distribution activities through a series of icons and dialogs. The top part of the screen holds the icons needed to initiate the activities, and the bottom part will display the type of process and the results of the process that is initiated.

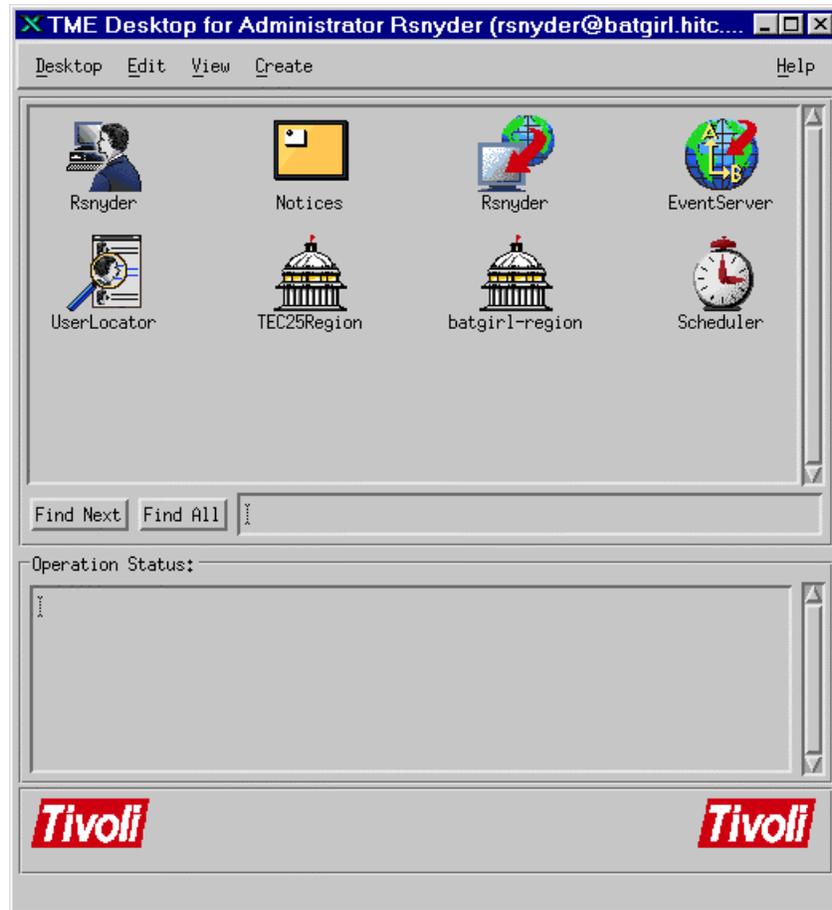


Figure 4.4.6-1. Tivoli Main Screen

4.4.6.3 Required Operating Environment

Tivoli/Admin and Tivoli Event Console require a UNIX platform that serves as the Tivoli Managed Region Server and as the Tivoli Event Server. The Tivoli Managed Nodes can be UNIX platforms or PC's. ECS Tivoli managed nodes consists of Sun hosts (Solaris 2.5.1 OS), Hewlett Packard hosts (HP 10.0.1 OS), and SGI (SGI Irix 6.2) hosts.

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled ReadMe file for each product. To find the ReadMe file for Tivoli/Admin and Tivoli/Event Console, use the XRP Baseline Manager to determine where in ClearCase the ReadMe file resides.

4.4.6.4 Databases

Tivoli uses two databases, a Sybase database and a Tivoli proprietary database. Operators are unable to access any of the information that either of them contain.

4.4.6.5 Special Constraints

The operator must be a Tivoli Administrator to create the regions and associated components.

4.4.6.6 Outputs

Output from Tivoli is displayed on the screen in the form of its GUIs or a log entry (discussed in Section 4.4.6.7).

4.4.6.7 Event and Error Messages

All event and summary error messages are displayed in the Operation Status box within the main screen (Figure 4.4.6-1). Certain error messages may also appear on the Xterm console that was used to start Tivoli.

4.4.6.8 Reports

None.

4.4.7 Cryptographic Management Interface (CMI)

The Cryptographic Management Interface (CMI) GUI program, *EcSeAuthnProg*, is used by operations personnel to generate a randomized username and password (though only the password is currently used) given a key. There is one key for each ECS server and is the same as the Program ID stored in a server's configuration file. This tool is most often used to generate passwords for Sybase and FTP user accounts. It is therefore recommended that access to this tool is restricted to Sybase and Unix System Administrators only.

CMI is used to perform the operator functions listed in Table 4.4.7-1.

Table 4.4.7-1. Common ECS Operator Functions Performed with CMI

Operating Function	Command / GUI	Description	When and Why to Use
Start <i>CMI</i> program.	<i>EcSeAuthnProg</i>	This will bring up the <i>ConnectAuth</i> GUI .	In order to obtain the user password for a given application key.
Generate password.	<i>CMI Main Screen (ConnectAuth GUI)</i>	This will cause the program to generate a randomized username and password.	This is only needed when a new user account is required by an ECS server.

4.4.7.1 Quick Start Using CMI

The CMI is a custom developed GUI utility and should be used only by operations personnel.

To execute CMI from the command line prompt, enter:

> **EcSeAuthnProg**

Refer to the 920-TDx-013 "Custom Code Configuration Parameters" documentation series for a listing of **EcSeAuthnProg**. **There are no configuration parameters for EcSeAuthnProg!**

4.4.7.2 CMI Main Screen

Figure 4.4.7-1 is the CMI GUI Screen which comes up when the CMI program is run. It contains three fields:

- Application Key field
- User Id field
- Password field

Operations personnel fill out the first field with the application key. In response, CMI will return a user name and password which are displayed in the associated fields.

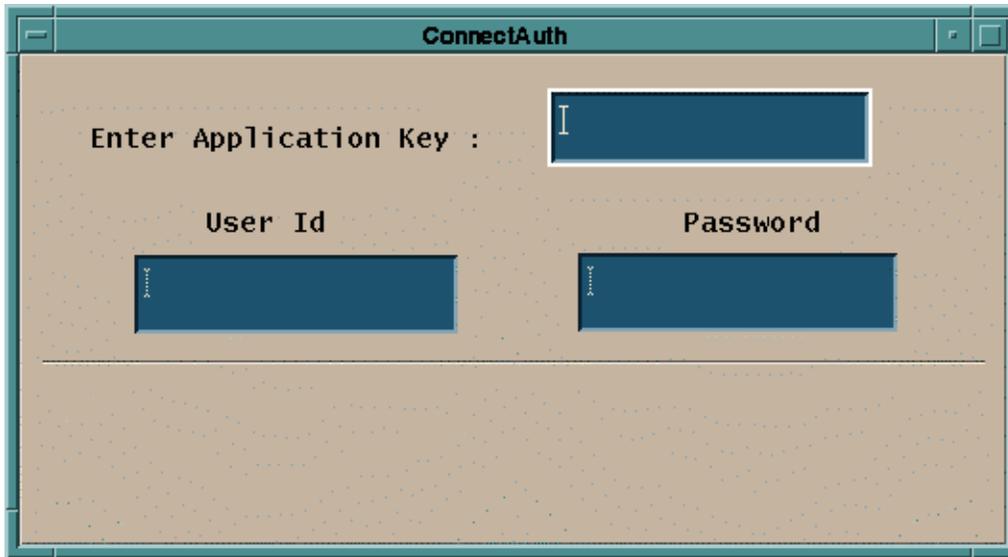


Figure 4.4.7-1. CMI Main Screen

The Table 4.4.7-2 below describes all the fields found in the CMI Screen in Figure 4.4.7-1.

Table 4.4.7-2. CMI Field Descriptions

Field Name	Data Type	Size	Entry	Description
Application Key	Integer	1 to 10 digits	Required	Key that identifies an application.
User Id	Character	8	Generated by <i>EcSeAuthnProg</i> program	Displays the randomized user id based on the key (this field is not used).
Password	Character	8	Generated by <i>EcSeAuthnProg</i> program	Displays the password to be used when creating the account.

4.4.7.3 Required Operating Environment

The *EcSeAuthnProg* depends on a data file which must be called “data” and must exist in the directory from which the tool is invoked. The data file is the same file as the *EcSeRandomDataFile* located in **\$ECS_HOME/<mode>/CUSTOM/security**, only with a different name. CMI requires no other configuration files. The program must be run on a Sun platform.

For further information on CMI refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC. Note, there are no configurable or tunable parameters or environment variables of CMI.

4.4.7.3.1 Interfaces and Data Types

CMI utilizes no special data types or interfaces.

4.4.7.4 Databases

None.

4.4.7.5 Special Constraints

A data file called “**data**” must exist in the execution directory. The data file must be the same file as the EcSeRandomDataFile.

4.4.7.6 Outputs

All information is displayed on the CMI screen.

4.4.7.7 Event and Error Messages

The CMI program issues error messages which are listed in Appendix A.

4.4.7.8 Reports

None.

This page intentionally left blank.

4.5 Science Software Integration and Test (SSI&T)

This section describes the tools used by DAAC operations personnel who are Science Software Integration and Test (SSI&T) specialists. The function of SSI&T is to prepare the science software received from the Instrument Teams for DAAC production. The tools that are available to the specialist cover a broad spectrum. Some are COTS tools that fall into the category of “Office Automation” such as spreadsheet and word processing packages. Others are COTS but are more specialized, such as graphic display packages. All the COTS tools/products are documented in separate product specific documentation. These tools are only identified in this section. Operators will have to verify that COTS documentation matches the product version in use. Finally, there are custom applications that are unique to the SSI&T activity. These tools are described in the following subsections.

This page intentionally left blank.

4.5.1 SSIT Manager

The SSIT Manager is the starting point for use by the SSI&T specialist to check in and verify the science software delivered by the instrument teams at the SCFs. The SSIT Manager application provides access to all COTS tools and custom applications that are part of the SSI&T environment. The SSIT Manager GUI is capable of kicking off instrument-specific compilation and execution scripts, configuration management scripts, custom code checking, file display and comparison tools, and COTS tools such as analysis environment programs. The SSIT Manager GUI contains a checklist of SSI&T steps in delivery and testing of science software, and a display of a log file recording SSI&T events. The checklist and log are the only inherent functionality to SSIT Manager; all other programs run from the Manager are also accessible from the Unix command line.

The terms Process Control File (PCF) and Object Description Language (ODL) are used in the following sections. The PCF is a file that tells an executable where to find its various inputs and outputs, as well as the values for any specific runtime parameters. Different various of these files are used both by the SSIT Manager and by PGEs. ODL is a parameter = value format for input files. It is used to define the PGEs to the Planning and Data Processing Database.

Table 4.5.1-1 presents a summary of the capabilities provided via the SSIT Manager GUI.

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (1 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Prepare SSI&T checklist.	SSIT Manager GUI	<ul style="list-style-type: none"> • SSIT Manager GUI requires a checklist which it can access • the checklist contains all the operational procedures for science software integration and test 	<ul style="list-style-type: none"> • during normal SSIT operations, used to keep track of activities pending and completed • checklist of operational procedures must first be prepared using a text editor. • SSIT Manager must then be linked to the checklist before invoked
Change SSI&T Checklist.	SSIT Manager GUI	<ul style="list-style-type: none"> • SSIT Manager GUI requires to change the checklist • the checklist contains all the operational procedures for science software integration and test 	<ul style="list-style-type: none"> • Update tracking of activities pending and completed • checklist of operational procedures must first be prepared using a text editor. • SSIT Manager must then be linked to the checklist before invoked

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (2 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
open xterm session.	<ul style="list-style-type: none"> • open xterm window via Tools pull-down menu • also can be opened via Unix command <i>xterm</i> 	Standard Unix command line window.	As needed for ad hoc use.
Code analysis.	select SPARCworks via Tools: Code Analysis pull-down menu	Used for ad hoc analysis of science software.	used to debug problems (e.g., memory leaks)
check for standards compliance.	select the following via Tools: Standards Checkers pull-down menu: <ul style="list-style-type: none"> • FORCHECK • Prohibited Function Checker • Process Control File (PCF) Checker • Prolog extractor 	<ul style="list-style-type: none"> • check FORTRAN 77 science software • check if certain functions are used in the science software which conflict with the production environment • check the syntax of the data in the Process Control File • Extract prologs from science software 	<ul style="list-style-type: none"> • to ensure that science code conforms to ECS standards • to ensure that the delivered PCF is of the proper syntax • To extract prologs from science software
Product Examination.	select the following via Tools:Product Examination pull-down menu: <ul style="list-style-type: none"> • IDL • EOSView 	Opened via Tools pull-down menu.	<ul style="list-style-type: none"> • Ad-hoc graphical analysis • For viewing an arbitrary file (e.g., standard product) in HDF format
File Comparison.	select the following via Tools: Product Examination File Comparison <ul style="list-style-type: none"> • ASCII files • Binary files • HDF files (GUI) • HDF files (hdiff) 	Compares the outputs of the science software between the DAAC and SCF.	Ensures that the output that was generated at the SCF when running the science software is the same output that is generated at the DAAC.

Table 4.5.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (3 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Edit text.	select the following via Tools: Text Editors pull-down menu: <ul style="list-style-type: none"> • Emacs • Xedit 	Text editors.	Edit arbitrary text file.
PDPS Database	select the following via Tools:PDPS Database pull-down menu: <ul style="list-style-type: none"> • PCF ODL Template • Check ODL Files • SSIT Science Metadata Update • SSIT Operational Metadata Update GUI 	<ul style="list-style-type: none"> • creates an ODL file template from the science software PCF • Check the ODL file updates PGE and ESDT SCIENCE metadata in the PDPS /SSIT database • updates PGE OPERATIONAL metadata via GUI in the PDPS /SSIT database 	To initialize and update the Planning/Production (PDPS) databases: <ul style="list-style-type: none"> • SSIT version • Production version
Data Server	select the following via Tools: Data Server pull-down menu: <ol style="list-style-type: none"> 1. Acquire DAP 2. Insert Static 3. Insert Test Dynamic 4. Insert EXE TAR 5. Edit SSAP 6. Get MCF 	<ol style="list-style-type: none"> 1. Acquires DAP. 2. Inserts static input file. 3. Inserts test dynamic input file. 4. Inserts tar file with files needed for Processing. 5. Edits Science Software Archive Package (SSAP) components. 6. Acquires Metadata Configuration Files from the Data Server. 	<ol style="list-style-type: none"> 1. After DAP notification received by email. 2. After ESDT is registered in Data Server, before test PGE run. 3. After ESDT is registered in Data Server, before test PGE run. 4. After PGE compilation, before test PGE run. 5. After PGE testing is complete, at time of promotion to Production, as needed to edit/review SSAP components. 6. As needed to retrieve MCF.

4.5.1.1 Quick Start Using SSIT Manager

The SSIT Manager provides a common interface to the SSI&T tools. An overview of the SSIT Manager GUI is provided in Section 4.5.1.2. A more detailed discussion of the tools accessed via this GUI is provided in subsequent sections.

The following assumptions are made with regard to the use of the SSIT Manager application.

- The operator is located at a workstation or server to which the SSIT Manager has been configured

- The operator has proper authorization to access the PDPS/SSIT database and the Data Server
- To access files in ClearCase, the operator has a ClearCase view already set
- The operator's environment has been configured as documented in the pertinent sections of the Help menu, available from the main window of the SSIT Manager (see Figure 4.5.1-1) The Index submenu of the Help menu provides access, through the Netscape browser, to a number of topics that help the operator in the environment configuration. Refer to section 4.5.1.2.10 for additional details on the Help Menu. A list of topics that can be searched through the Help menu is shown in Figure 4.5.1-36.

To start SSIT Manager at the Unix command line:

```
DpAtMgr ConfigFile <config_filename> ecs_mode <mode> &
```

where:

config_filename is the name of the user's personal Process Framework configuration file for this program, as customized from **\$ECS_HOME/CUSTOM/cfg/**.

ecs_mode is the ECS mode of operation, e.g., OPS, TS2

4.5.1.1.1 Invoking SSIT Manager From the Command Line Interface

To start SSIT Manager at the Unix command line:

```
DpAtMgr ConfigFile <config_filename> ecs_mode <MODE> &
```

where config_filename is the name of the user's personal Process Framework configuration file for this program, as customized from **\$ECS_HOME/CUSTOM/cfg/**.

4.5.1.1.1.1 Sun Platform

Table 4.5.1-2 lists the SSI&T command line interfaces for the Sun workstation.

Table 4.5.1-2. Command Line Interfaces (Sun) (1 of 3)

Command Line Interface	Description and Format	When and Why Used
EcDpAtMgr	Startup of SSIT Manager.	To do SSI&T, and record items accomplished in the log.
EcDpAtMgrLogDump	Used to dump/print a log file to the screen.	As needed.
xterm	Open a Unix command line window.	As needed.

Table 4.5.1-2. Command Line Interfaces (Sun) (2 of 3)

Command Line Interface	Description and Format	When and Why Used
sparcworks	Ad hoc code analysis	As needed.
ghostview	Postscript file viewer	As needed.
netscape	WWW browser Netscape	As needed.
acroread	PDF file viewer Adobe Acrobat	As needed.
DpAtMgrForcheck	FORTRAN 77 code analysis	Determine whether FORTRAN 77 science software adheres to standards.
EcDpAtBadFuncGui	Prohibited function checker (GUI)	Determine whether science software adheres to standards.
EcDpAtBadFunc	Prohibited function checker (command line)	Determine whether science software adheres to standards.
EcDpAtCheckPCF	Process Control File checker (GUI)	Determine whether PCF is valid.
EcDpAtMgrPrologs	Prolog extractor	Extract science software code prologs.
/data/IDL/idl_4/bin/idl	IDL	As needed.
EOSView	EOSView	HDF file viewer.
EcDpAtMgrXdif	ASCII file comparison	Compare 2 text files.
EcDpAtBinDiffGui	Binary file difference environment	Compare 2 binary files.
DpAtCheckHdfFile	HDF file comparison (GUI)	Compare 2 HDF files.
DpAtHdif	HDF file comparison (command line)	Compare 2 HDF files.
xedit	Text editor	As needed.
emacs	Text editor	As needed.
EcDpAt CreateODLtmplate	Create PGE metadata ODL template file.	Before running EcDpAtDefinePGE .
EcDpAtVerifyODL	Verify PGE metadata ODL template file.	Before Running EcDpAtDefinePGE.
EcDpAtDefinePGE	Update PDPS/SSIT database with SCIENCE metadata.	Before executing PGE in SSIT environment.
DpAtOpDbGui	Update PDPS/SSIT database with OPERATIONAL metadata.	Before executing PGE in SSIT environment.
EcDpAtStageDAP	Acquires DAP from the Data Server.	After email subscription notification received.
DpAtInsertStatic	Inserts static input file into the Data Server.	Before executing PGE in SSIT or Production environment.
DpAtInsertTest	Inserts test dynamic input file into the Data Server.	Before executing PGE in SSIT environment.

Table 4.5.1-2. Command Line Interfaces (Sun) (3 of 3)

Command Line Interface	Description and Format	When and Why Used
DpAtInsertExeTar	Inserts tar file of executables, etc. needed to run PGE file into the Data Server.	Before executing PGE in SSIT or Production environment.
EcDpAtSSAPGui	Edit and inserts a single SSAP component into the Data Server.	After SSI&T is finished, before official promotion to Production.
netscape <html page name>	HTML pages for acquiring SSAP components from the Data Server, including test outputs.	During SSI&T, to get test outputs; After SSI&T is finished.
EcDpAtaCQUIREMCF	Get ESDT from the Data Server and insert MCF.	Before inserting MCF in the Data Server.

4.5.1.1.1.2 SGI Platform

It is intended that the SSI&T tools be most often run from the SSIT Manager. A small number of SSIT tools run only on the SGI platform. Because of security considerations, these tools cannot be run from the SSIT Manager on the Sun. They may only be run from the Unix command line on the SGI platform as indicated in Table 4.5.1-3.

Table 4.5.1-3. Command Line Interfaces (SGI)

Command Line Interface	Description and Format	When and Why Used
usr/sbin/cvproj	ProDev Workshop: Used for ad hoc analysis of science software.	Used to determine causes of problems (e.g., memory leaks).
DpAtRusage	Measures PGE performance.	Output of this tool is to be typed into the "Performance Statistics" section of the PROFILE screen of the PDPS/SSIT Database Update GUI.

Table 4.5.1-4 lists SGI platform tools associated with the SSIT process.

Table 4.5.1-4. SGI Tools Description

Categories /Tools	Tool Description & Use	Further Information
ProDev Workshop	<ul style="list-style-type: none"> • ProDev Workshop is a COTS package developed by SGI • This tool is targeted within ECS for applications running on the SGI science processors • ProDev Workshop is a software development support tool which includes several tools that may have applicability to SSIT • Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks 	<ul style="list-style-type: none"> • ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSIT menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
PGE Performance	<ul style="list-style-type: none"> • DpAtRusage is a custom tool developed by ECS • It measures performance parameters such as CPU time used for a PGE linked to the SDP Toolkit, SCF version. 	A help message is printed if the tool is invoked without input parameters.

4.5.1.2 SSIT Manager

Figure 4.5.1-1 illustrates the SSIT Manager. This screen is the starting point for SSI&T activities. It provides access to a collection of tools that are used in the SSIT process. The procedures for SSI&T using these tools are defined in other documentation in conjunction with the DAAC SSI&T personnel and the instrument teams.

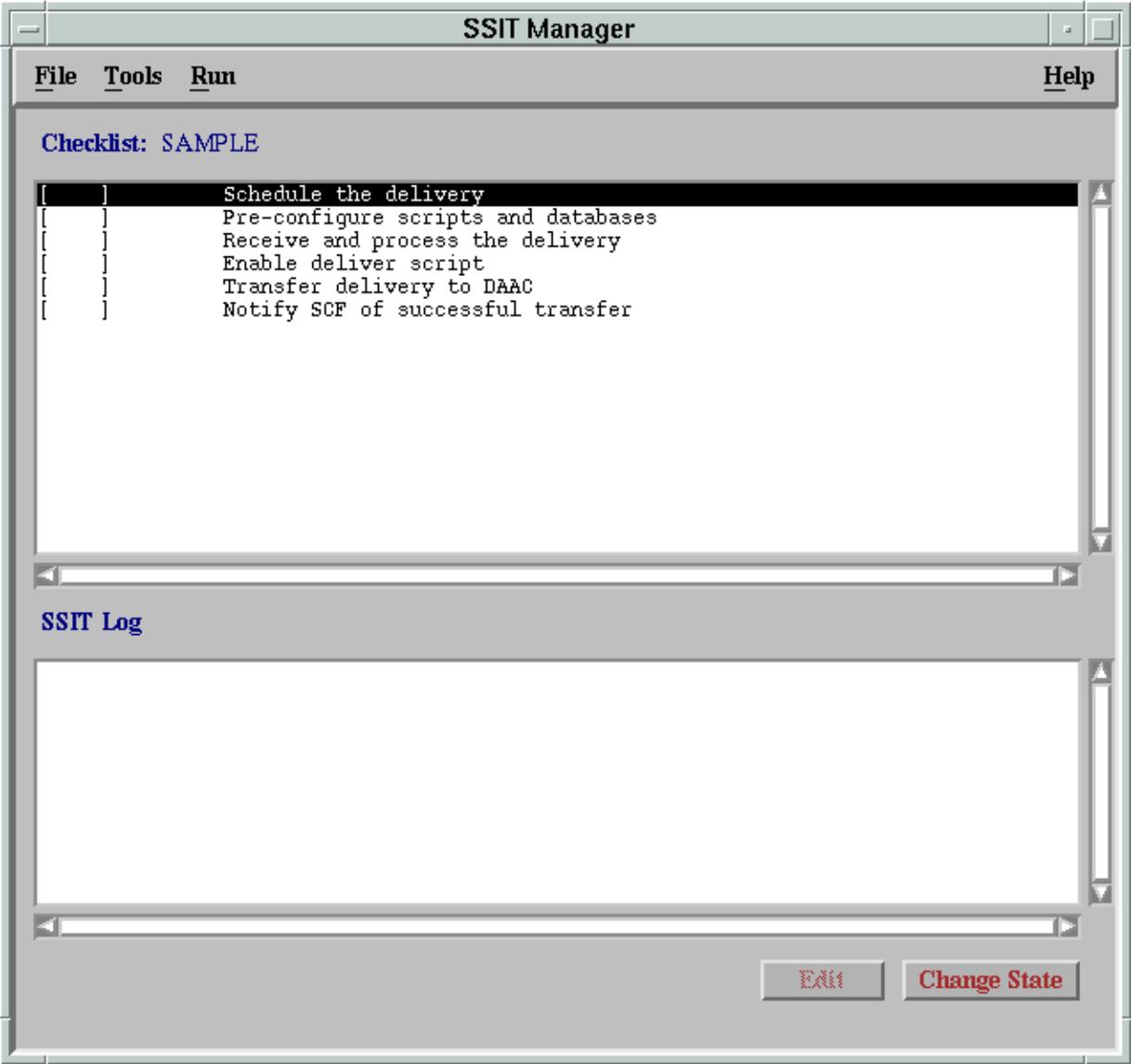


Figure 4.5.1-1. SSIT Manager Main Screen

The SSIT Manager Window includes two windows of information, a menu bar at the top of the screen, and two buttons at the bottom of the window.

The Checklist Window

The top window is the SSI&T Checklist. This lists a configurable set of steps to be completed during the SSI&T process. It is displayed as a line of text for each item, with a check box that is checked or unchecked. A checklist may be tailored specifically to a science software suite being delivered. The checklist is manual, in that the operator must check it with a mouse click. The checklist is configurable by editing a checklist text file as explained under the *Checklist* topic of Help/Index menu.

The SSIT Log Window

The lower window is a log of activities accomplished in the SSI&T process. As steps in the process are accomplished, a log entry is automatically added. Each time a box on the checklist is checked or unchecked, the log is updated. The user is required to annotate any changes.

The Menu Bar

The menu bar includes a simple selection of options:

- **File** - Offers the option to exit from the SSIT Manager application.
- **Tools** - Provides access to a collection of tools that are of use in the SSI&T process. The tools that are available are described in the following section.
- **Run** - User-customizable menu, intended for programs such as scripts to run science software.
- **Help** - Provides context-sensitive help.

SSIT Manager Buttons

There are also two buttons at the bottom of the screen.

- **Edit** - The checklist items may be edited by the SSI&T specialist to include additional detail on the SSI&T process. Clicking the Edit button brings up the text editor for this purpose.
- **Change State** - This toggle button allows the SSI&T specialist to switch between “checked” and “unchecked” for the selected checklist item.

4.5.1.2.1 SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager. After selecting the Tools menu option of the menu bar, a set of options is available(see Figure 4.5.1-2 which shows the items available from the menu item).

The first item on the list, the Xterm option, will start an Xterm window session. Selecting any of the other options displays a secondary list of particular tools within that category. The tools categories and tools are described in the following table.

Many of the items accessed through the Tool Menu are COTS products for which documentation exists and requires no further discussion. Some items are custom applications from other ECS subsystems (e.g., EOSView); the complete description and documentation for these items is

provided in other parts of this document. The remaining items are custom applications and are described in the following subsections.

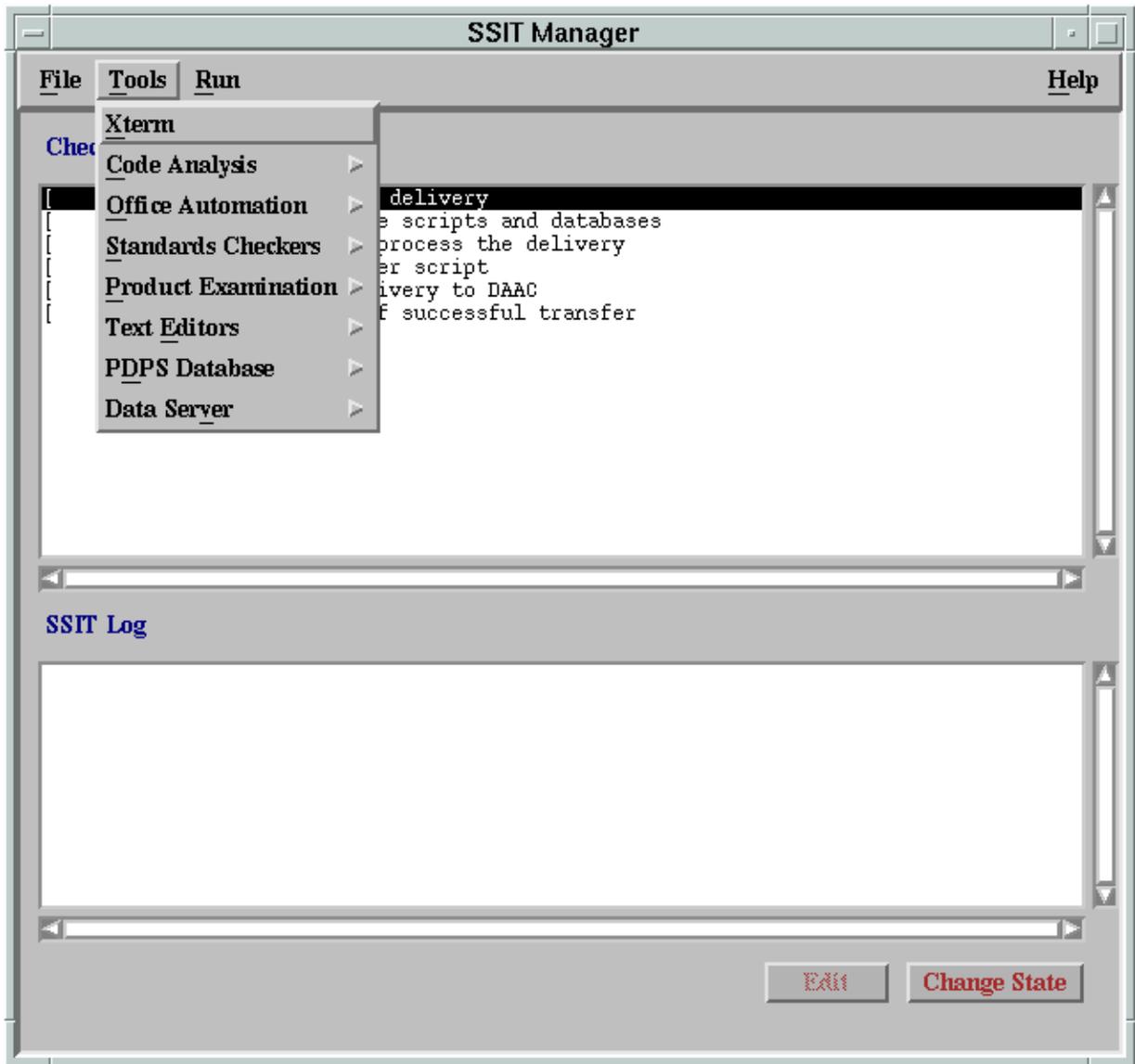


Figure 4.5.1-2. SSIT Manager Main Screen - Tools Menu

Table 4.5.1-5 describes the SSIT Manager Tools.

Table 4.5.1-5. SSIT Manager Tools Description (1 of 5)

Category	Tools	Tool Description & Use	Further Information
UNIX Access	Xterm	<p>This tools option allows the operator to open an Xterm window and enter Unix commands, as needed.</p> <p>The other SSIT Manager windows remain available while an Xterm window is open.</p>	<p>Unix documentation is available online via 'man' command.</p>
Code Analysis	SPARCworks	<ul style="list-style-type: none"> • SPARCworks is a COTS package developed by Sun Micro Systems • Selecting this option from the Tools list will start the SPARCworks tool which opens a separate window • Among the tools that are available in SPARCworks is the code analysis tool which allows for static code analysis to aid in detecting memory leaks • For the SGI systems the ProDev Workshop product provides similar functionality. 	<ul style="list-style-type: none"> • SPARCworks includes online documentation describing its features • Other SPARCworks documentation is delivered with ECS. This documentation includes: <ul style="list-style-type: none"> – SPARCworks Browsing Source Code – SPARCworks Debugging a Program 3.0.1 – SPARCworks Tutorial 3.0.1 – SPARCworks Performance Tuning an Application 3.0.1
Office Automation	GhostView	<ul style="list-style-type: none"> • Selecting Ghostview from the Tools menu item starts the Ghostview application • Ghostview is a freeware tool • This is used for displaying Postscript format files 	<ul style="list-style-type: none"> • Ghostview includes online documentation on its use • Since this is a freeware product, there is no other documentation provided by ECS for Ghostview
	Netscape	<ul style="list-style-type: none"> • Netscape is a COTS package developed by Netscape Corp. • This tool is a 'Browser' which is used to access Internet WWW web sites • EOSDIS has adopted the WWW and Netscape as an internal standard for the publication and exchange of various types of documentation, schedules, etc. • ECS documentation is available through the WWW, including documentation related to SSIT 	<ul style="list-style-type: none"> • Documentation on Netscape is available online • Other documentation on Netscape is available in this document in Section 4.12.5, "Netscape Navigator."

Table 4.5.1-5. SSIT Manager Tools Description (2 of 5)

Category	Tools	Tool Description & Use	Further Information
	Adobe Acrobat	<ul style="list-style-type: none"> • Adobe Acrobat Reader is a COTS package developed by Adobe Systems Inc. • This product is used to view electronic documents that have been created in the Portable Document Format (PDF) file format • Many of the ECS documents have been created or converted to this format 	<ul style="list-style-type: none"> • Documentation on Adobe Acrobat products is available online • Other documentation on Adobe Acrobat is available in this document in Section 4.12.5, "Netscape Navigator" • Other documentation for this application is delivered with ECS
Standards Checkers	FORCHECK	<ul style="list-style-type: none"> • FORCHECK is a COTS tool used to verify that software follows certain coding standards • FORCHECK will analyze source code that is written in Fortran 77 with ANSI extensions • When selected from the SSIT Manager window, and xterm window is created in which the program runs • There is no GUI associated with FORCHECK. • Note that FORCHECK is configured to check for the non-ANSI FORTRAN 77 extensions which are approved for use in ECS science code, according to "Data Production and SCF Standards and Guidelines." 	<ul style="list-style-type: none"> • Documentation on FORCHECK is available online • Other documentation for this applications is delivered with ECS. This includes: • FORCHECK for Sun/SunOS, A Fortran Verifier and Programming Aid, User Guide
	Prohibited Function Checker	<ul style="list-style-type: none"> • The Prohibited Function Checker is a custom application used to determine if the science software has been developed using coding standards established by the ESDIS Project ("Data Production and SCF Standards and Guidelines") in conjunction with the instrument teams responsible for developing the software • These coding standards are limited to prohibitions against the use of certain types of functions that may cause problems in the production environment • The Prohibited Function Checker checks for functions defined in a science software source code. 	<ul style="list-style-type: none"> • On-line help is provided with the Prohibited Function Checker • A description of the use of the tool is provided in the following sections. <p>This tool is described in detail in the following 4.5.1.2.4.2 section.</p>

Table 4.5.1-5. SSIT Manager Tools Description (3 of 5)

Category	Tools	Tool Description & Use	Further Information
Standards Checkers	Process Control File Checker	<ul style="list-style-type: none"> • The Process Control File (PCF) checker is a custom application used to verify that the format of the PCF is correct • A PCF is a part of the delivered science software delivery package • The PCF specifies the names and locations of files used by the science software executables, and defines the correspondence between (1) the file specifications and (2) the logical identifiers used by the science software to reference the specified files. 	<ul style="list-style-type: none"> • On-line help is provided with the PCF Checker • A description of the use of the tool is provided in the following sections • Note that this tool consists of a GUI front end added to an ECS SDP Toolkit utility • Documentation of the utility is found in the SDP Toolkit Users Guide for the ECS Project. <p>This tool is described in detail in the following 4.5.1.2.4.3.1</p>
	Prolog extractor	Custom application which extracts code prologs from science software.	Online help is provided
Product Examination	IDL	<ul style="list-style-type: none"> • IDL (Interactive Data Language) is a COTS product, developed by Research Systems Inc. • It is used to interactively visualize and analyze scientific and engineering data products • IDL can be used with data files of an arbitrary format. 	<ul style="list-style-type: none"> • Documentation on IDL is available online • Other documentation for this applications is delivered with ECS
	EOSView	<ul style="list-style-type: none"> • EOSView is a custom application used to display HDF format files and the metadata associated with those files • EOSView is expected to be primarily by the EOS scientific community, but will also be used by the operations community to support the SSIT process (for file inspection, etc.) and by the DAAC quality assurance groups to view and analyze problems arising from production operations 	<ul style="list-style-type: none"> • Documentation on EOSView is available online • EOSView is integral with the ECS Desktop; for more information, see Section 4.12.7, "EOSView"

Table 4.5.1-5. SSIT Manager Tools Description (4 of 5)

Category	Tools	Tool Description & Use	Further Information
Product Examination	File Comparison	<ul style="list-style-type: none"> • File comparison tools, which are custom products of ECS, are provided with the SSIT Manager • Tools are provided to support file comparison in three different formats: ASCII, Binary, and HDF • These tools would be used by the SSIT Specialists to compare the benchmark data products generated at the SCFs with the data products generated as a part of the SSIT process at the DAAC • All standard data products will be produced in HDF format, for which the HDF file comparison utility will be used • For intermediate or other data products that may not adhere to the standards, the binary or ASCII file comparison tools may be used • In the case of binary files, the DAAC operator must write code to do the comparison, for which a support environment is provided 	<ul style="list-style-type: none"> • Documentation on the file comparison tools is available online • These tools are described more fully in following subsections of this document.
Text Editors	Emacs	Emacs is a freeware text editing tool most frequently used by the SSIT specialists in developing scripts or ad hoc programs to be run in the SSIT environment.	<ul style="list-style-type: none"> • Documentation on Emacs is available online • Since this is a freeware product, there is no other documentation provided by ECS
	Xedit	Xedit is a freeware text editing tool most frequently used by the SSIT specialists in developing scripts or ad hoc programs to be run in the SSIT environment.	<ul style="list-style-type: none"> • Xedit includes online documentation on its use • Since this is a freeware product, there is no other documentation provided by ECS
PDPS Database	PCF ODL Template	Creates a PDPS PGE SCIENCE metadata ODL file from a science software PCF.	This tool is described in detail in the following 4.5.1.2.8.1 section.
	Check ODL	Parses ODL file and records errors. This tool accesses the PDPS DB for information on PGE/ESDT whose ODL is already available.	This tool is described in detail in the following 4.5.1.2.8.2 section.
	SSIT Science Metadata Update	Updates PDPS/SSIT database with PGE and ESDT SCIENCE metadata read from ODL files.	This tool is described in detail in the following 4.5.1.2.8.3 section.

Table 4.5.1-5. SSIT Manager Tools Description (5 of 5)

Category	Tools	Tool Description & Use	Further Information
	SSIT Operational Metadata Update GUI	Updates PDPS/SSIT database with PGE OPERATIONAL metadata input by user.	This tool is described in detail in the following 4.5.1.2.8.4 section.
Data Server	Acquire DAP	Acquires a DAP from the Data Server.	This tool is described in detail in the following 4.5.1.2.9.1 section.
	Get MCF	Gets information about the ESDT the MCF is acquired for and interface with the Data Server to get the desired MCF.	This tool is described in detail in the following 4.5.1.2.9.2 section.
	Insert Static	Inserts a static input file to the Data Server.	This tool is described in detail in the following 4.5.1.2.9.3 section.
	Insert Test Dynamic	Inserts a test dynamic input file to the Data Server.	This tool is described in detail in the following 4.5.1.2.9.4 section.
	Insert EXE TAR	Inserts tar file of executables, etc. needed to run science software to the Data Server.	This tool is described in detail in the following 4.5.1.2.9.5 section.
	SSAP Editor	Read/Write Metadata and record errors.	This tool is described in detail in the following 4.5.1.2.9.6 section.

4.5.1.2.2 Xterm

Xterm is a COTS product that activates an Xterm window. This allows the operator to use the Unix environment and resources from a PC. The Xterm window as shown in Figure 4.5.1-3 has all the properties of a PC window while the “screen” portion is essentially a Unix terminal. The user will need environment parameters for the PC, such as the IP Address, to use Xterm.

There is no submenu for the Xterm menu.



Figure 4.5.1-3. Xterm Unix Terminal Window

4.5.1.2.3 Code Analysis

The Code Analysis menu as shown in Figure 4.5.1-4 provides the SPARCworks tool to assist in analyzing C code.



Figure 4.5.1-4. SSIT Manager Main Screen with Code Analysis tool menu displayed.

4.5.1.2.3.1 SPARCworks

SPARCworks is a COTS product. This function is not documented for operators. Analysis of the program code in the operational environment will result in significant performance reduction.

4.5.1.2.4 Office Automation

The Office Automation menu as shown in Figure 4.5.1-5 provides tools to assist operators in documentation and administrative tasks.

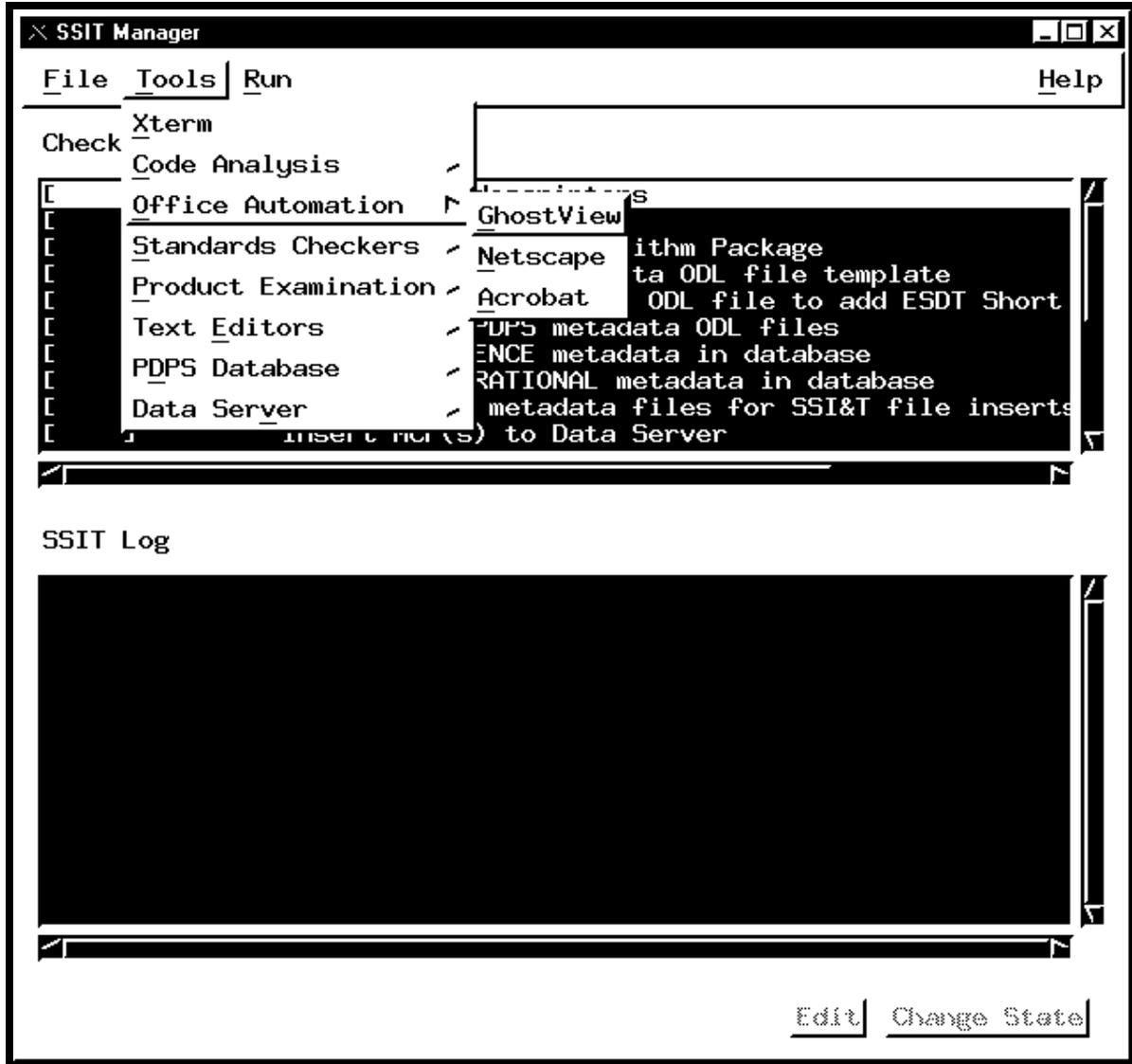


Figure 4.5.1-5. SSIT Manager Main Screen with Office Automation tool menu displayed.

4.5.1.2.4.1 GhostView

GhostView is a COTS product. GhostView as shown in Figure 4.5.1-6 provides the operator with a tool for viewing and editing graphics files. The popular PostScript format (.ps) is used to capture screen images and other pictures used in documentation. The operation of this tool follows standard PC application rules. Images displayed in GhostView may be cut & pasted into MS Word® documents.

More information about GhostView is available on the Web free from
<ftp://prep.ai.mit.edu/pub/gnu/>

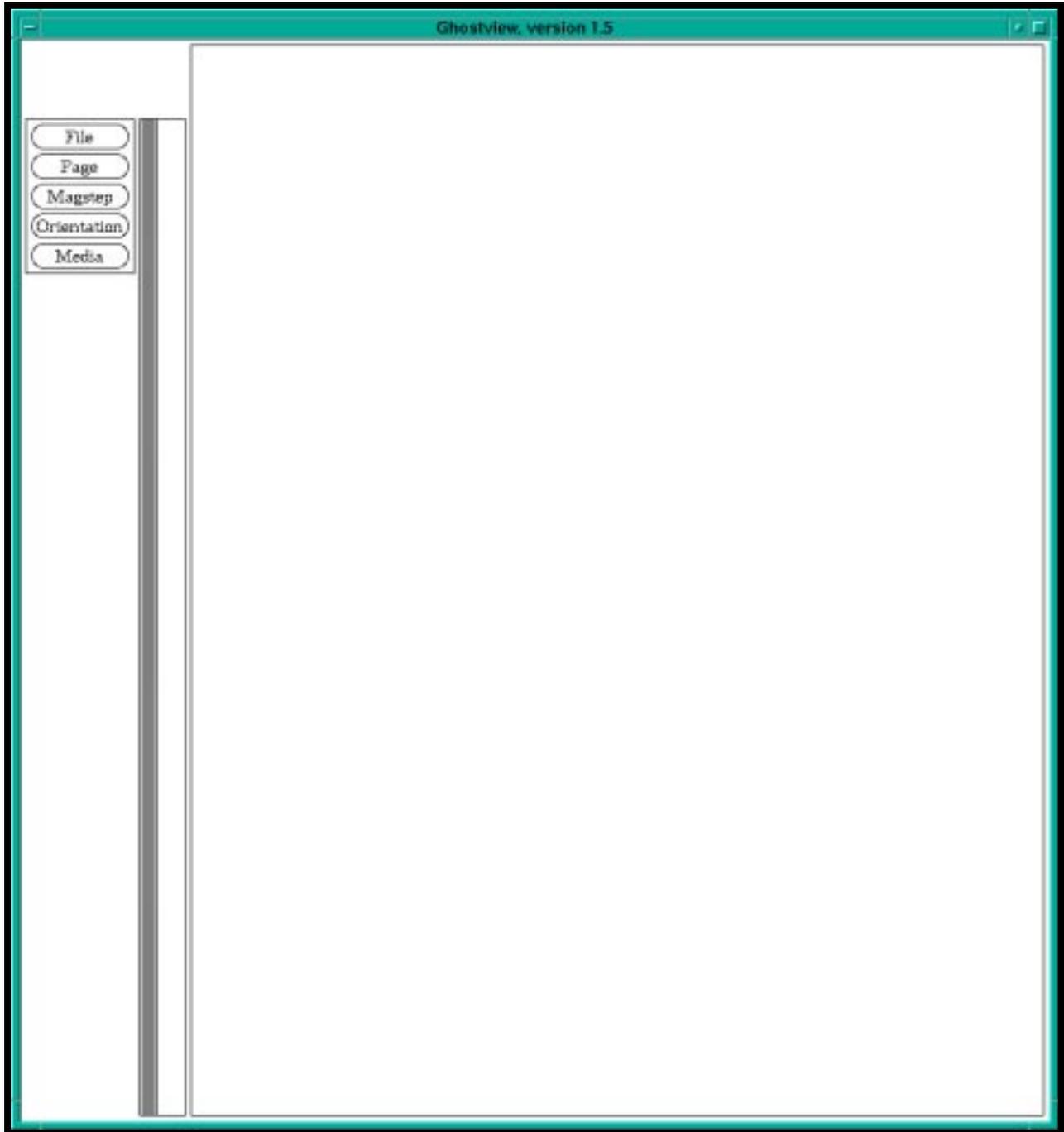


Figure 4.5.1-6. GhostView Main Screen

4.5.1.2.4.2 Netscape

Netscape is a COTS product. Netscape is the project Internet Browser, it is described in section 4.12.3, “Netscape Communicator”.

4.5.1.2.4.3 Acrobat

Acrobat is a COTS product. Adobe Acrobat is a documentation package oriented to storing data as graphics files. The main window is shown in Figure 4.5.1-7. The package is geared to fast printing by not requiring translation of text files to printer format. The popular Acrobat (.pdf) file format is native to this package. Images or text displayed in Acrobat may be cut & pasted into Word® documents.

More information about Acrobat is available on the Web free from <http://www.adobe.com/>.

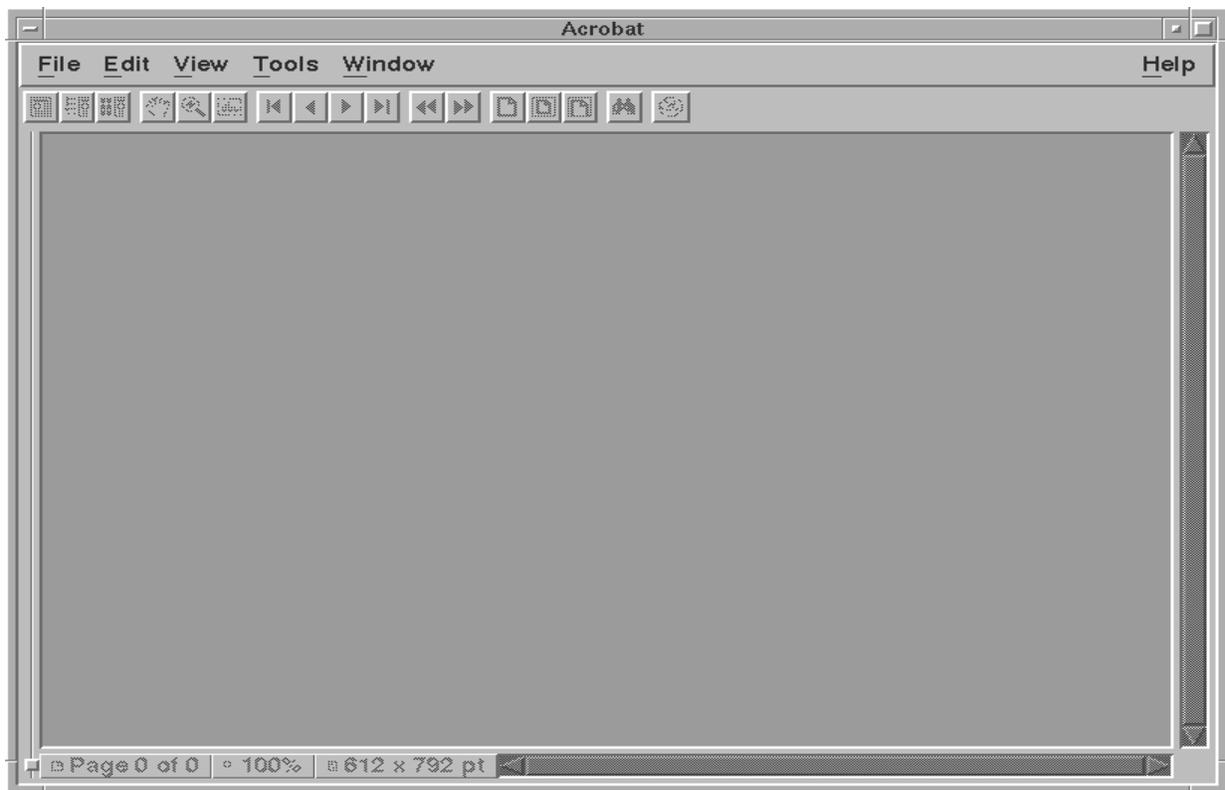


Figure 4.5.1-7. Acrobat Main Screen

4.5.1.2.5 Standards Checkers

The Standards Checkers menu as shown in Figure 4.5.1-8 provides tools to assist operators in validation of DAPs.

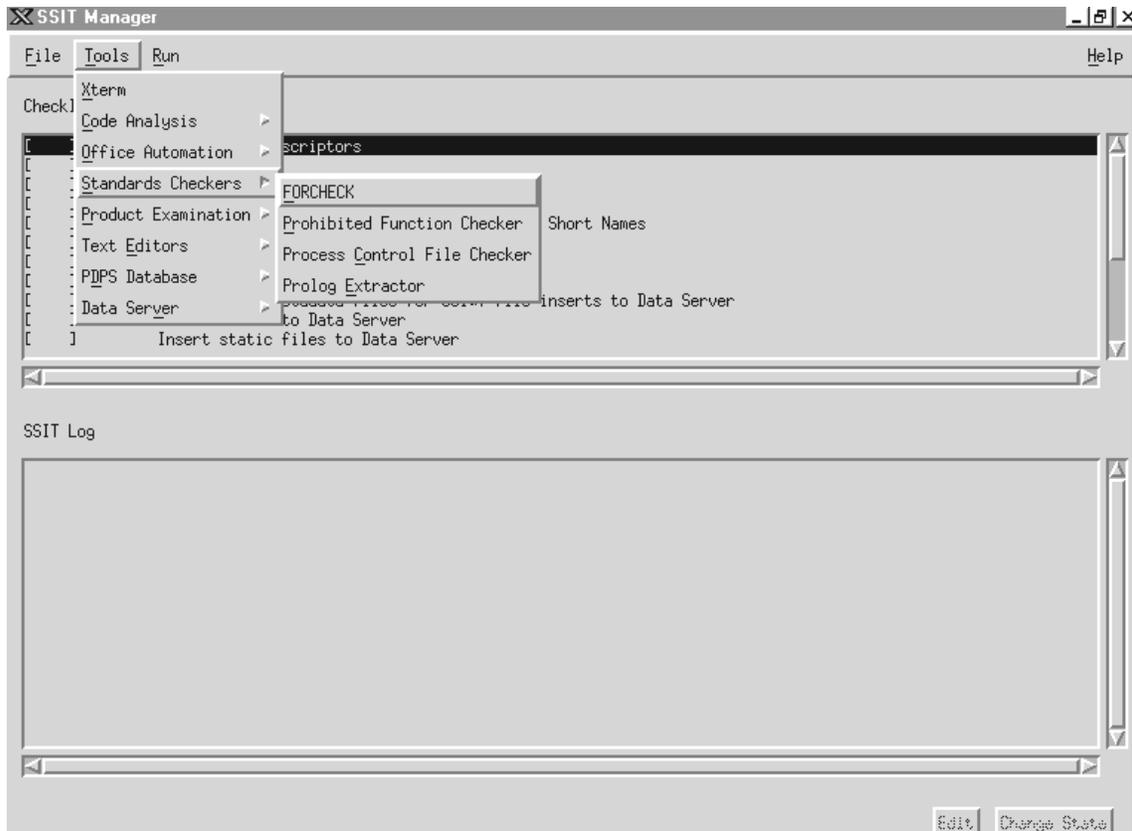


Figure 4.5.1-8. SSIT Manager, Tools Menu, Standards Checker Submenu Choices

4.5.1.2.5.1 FORCHECK

FORCHECK is a COTS tool. FORCHECK, shown in figure 4.5.1-9, checks science software written in FORTRAN 77. FORCHECK may be configured to determine whether the ANSI FORTRAN 77 extensions used in science software are allowed by the official NASA/ESDIS standard. The NASA/ESDIS standards are implemented through use of a FORCHECK configuration file, which is pointed to by the \$FCKCNF parameter. Note: FORCHECK does not process FORTRAN 90 code.

FORCHECK does not have a GUI, it is command line driven. The FORCHECK interface is via a spawned Xterm window.



Figure 4.5.1-9. FORCHECK program running in an Xterm window

When invoked, FORCHECK produces a query prompt to solicit changes to the default global options and the list (output) file. The default output file is the UNIX **stout** file, normally configured as the Xterm window. FORCHECK provides numerous Global and Local options to tailor the code analysis and the resulting output. See the *FORCHECK for Sun/SunOS A Fortran Verifier and Programming Aid version 12 User's Guide* for more information.

4.5.1.2.5.2 Prohibited Function Checker

The Prohibited Function Checker is an SSI&T tool that uses a GUI interface to check specified source code for prohibited function calls. The document, *Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, Revision A, October 1996 (423-16-01)*, mandates that science software delivered to the DAACs for integration into the ECS should be free of prohibited functions, which may cause errors in the production environment.

Prohibited Function Checker (PFC) GUI serves the following functionality:

- Toolkit to process program and text files for prohibited functions.
- GUI interface that allows the user to select file(s) to be checked.
- Writes output to a log.
- User can scan through any prohibited functions found from the output.

Table 4.5.1-6 provides the language and the file extensions that can be checked for prohibited function calls in the PFC Tool.

Table 4.5.1-6. File Name Extensions

Language	File Name Extensions
Ada	.a, .ada
C	.c, .h
C++	.cc
Fortran 77	.f, .f77, .ftn, .inc
Fortran 90	.f90
C shell	.csh
Korn shell	.kh
Bourne Shell	.sh
Perl	.pl
Text files	.txt

Note: Since SSI&T is a collection of various tools, there is no specific order in which they must be run. Most tools can be brought up from the SSI&T Manager Main Screen as well as can be started on their own. The File menu provides the capability to exit the manager, and the Tools menu gives access to various tools that make up SSI&T.

4.5.1.2.5.2.1 Checking for Prohibited Functions: GUI Version

Assumptions:

The SSI&T Manager is running.

Steps to Check for Prohibited Functions:

1. From the *SSI&T Manager Main Screen* (Figure 4.5.1-8), click on the **T**ools menu, then choose **S**tandards Checkers, and then choose **P**rohibited Function Checker to bring up the screen shown in Figure 4.5.1-10
 - This displays the *Prohibited Function Checker Pop-up*
 - This Pop-up has five buttons (Analyze, View, Report, Quit and Help) which are described in this section.

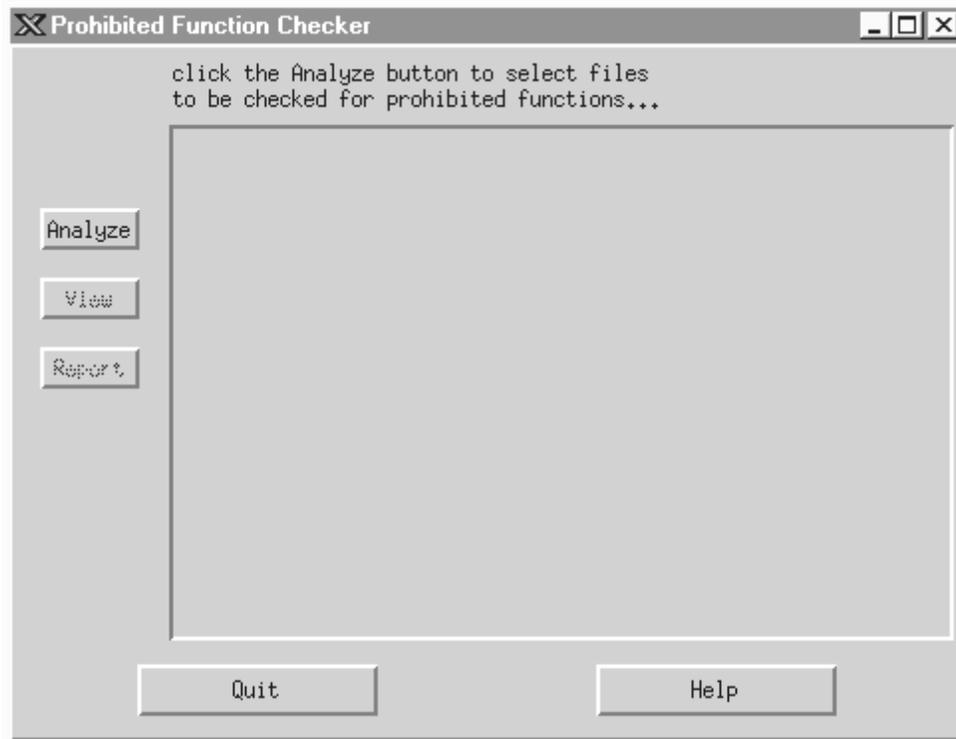


Figure 4.5.1-10. Prohibited Function Checker Pop-up

2. In the *Prohibited Function Checker POP-UP*, click on the **Analyze** button.
 - ◆ The *File Selector POP-UP* is displayed.
3. In the *File Selector POP-UP* (Figure 4.5.1-11 below), choose the **Directories** sub-window, and double click on the desired directory.

Repeat this step until the directory with the source files to be checked are displayed in the **Files** sub-window.
4. Within the **Files** sub-window, click on the source files to be analyzed. Highlight each source file by clicking on it.
 - ◆ To choose groups of contiguous files, hold down the left mouse button and drag the mouse down.
 - ◆ To choose non-contiguous files, hold down the Control key while clicking on file names.
5. The files selected will show in the **Selections** sub-window.

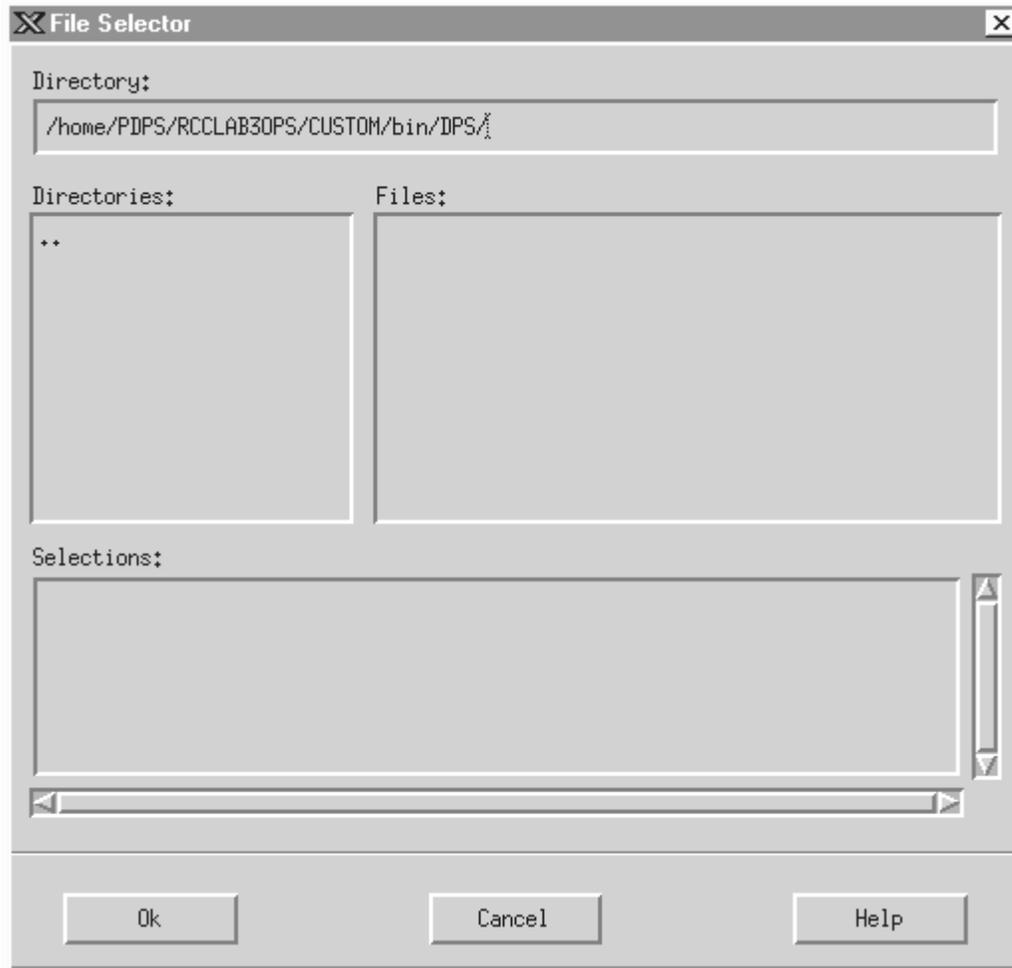


Figure 4.5.1-11. File Selector POP-UP

6. In the *File Selector POP-UP*, click on the **OK** button. The *Prohibited Function Checker* will then start analyzing for prohibited functions within the selected files.
7. The *File Selector POP-UP* disappears, and the *Prohibited Function Checker* window displays all of the selected files containing prohibited functions. The results of the *Prohibited Function Checker* can be examined using the View button (only one source file can be selected at a time to view the code) or Report button.
8. Highlight one of the source files in the *Prohibited Function Checker* window, and then click on the **View** button.
 - ◆ The *Source Code POP-UP* is displayed (Figure 4.5.1-12).
 - ◆ Occurrences of prohibited functions found in the source code are highlighted.

- ◆ Click on the **Next** button to bring into the window successive occurrences of prohibited functions within the same source file.
- ◆ Click on the **Done** button to close the *Source Code POP-UP*.
- ◆ All prohibited functions can be viewed similarly, by examining one source file at a time.

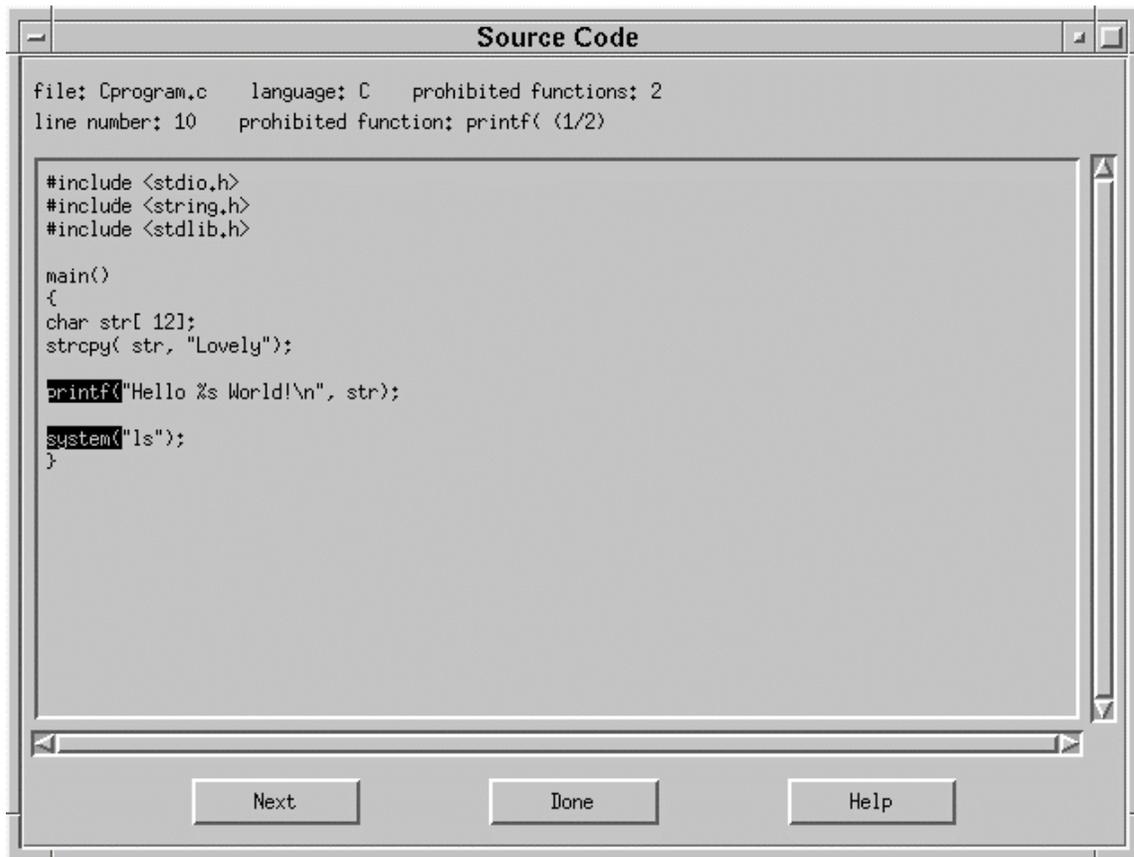


Figure 4.5.1-12. Source Code POP-UP

9. Use the Report button on the *Prohibited Function Checker POP-UP* to:
 - ◆ Display the *Report POP-UP* (Figure 4.5.1-13).
 - ◆ Display for each source file, a list of prohibited functions found.

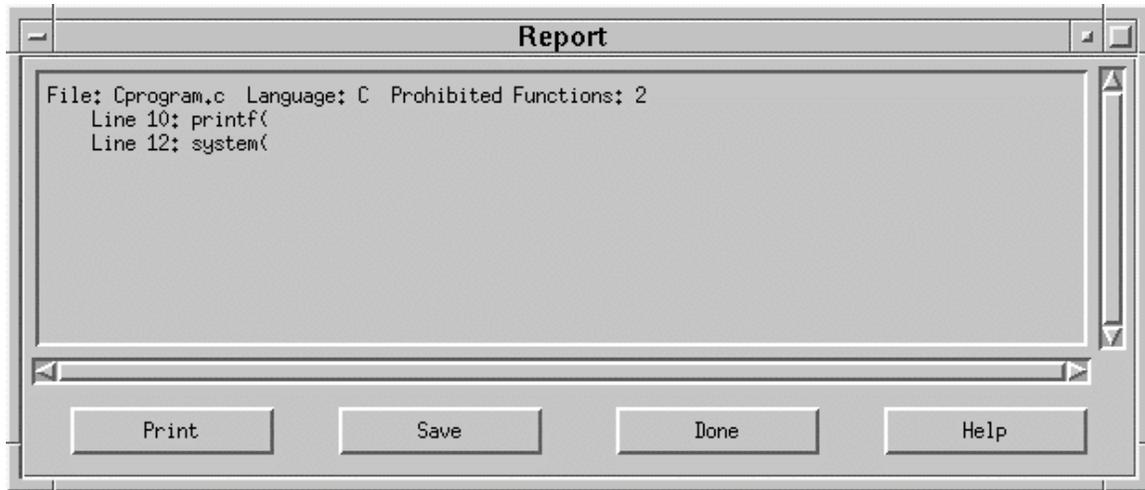


Figure 4.5.1-13. Report POP-UP

10. The results of the source file, showing prohibited functions can be saved or printed from the *Report POP-UP*.
 - ◆ Clicking on the **Save** button brings up the *Save To File POP-UP*. Specify the directory and file name in which to save the results.
 - ◆ Clicking on the **Print** button prints the results on the default printer.
11. The **Help** button within any WINDOW brings up the HELP POP-UP.

4.5.1.2.5.3 The Process Control File Checker

This section describes how the Process Control File (PCF) Checker tool is used by the SSI&T Specialist during the SSI&T process.

The PCF Checker verifies that the PCF provided by the Science Software development teams, as part of the DAP (Delivery Archive Package), is syntactically correct.

Before this program is used, the following condition must be met:

The PCFs to be checked are available, accessible, and have read permission for the operator.

4.5.1.2.5.3.1 Process Control File Checker

The PCF Checker POP-UP allows the user to select a PCF and have it scanned by Science Data Processing (SDP) Toolkit routines for any irregularities such as format or structure.

4.5.1.2.5.3.2 Quick Start Using This Tool

The PCF Checker tool is started from the TOOLS item on the menu bar, on the *STANDARDS CHECKER* menu from the SSIT Manager Pop-up. Figure 4.5.1-8 shows the submenu choices.

When the PCF Checker application is started from the TOOLS/STANDARDS CHECKER menu item, the Pop-up shown in Figure 4.5.1-14 is displayed.

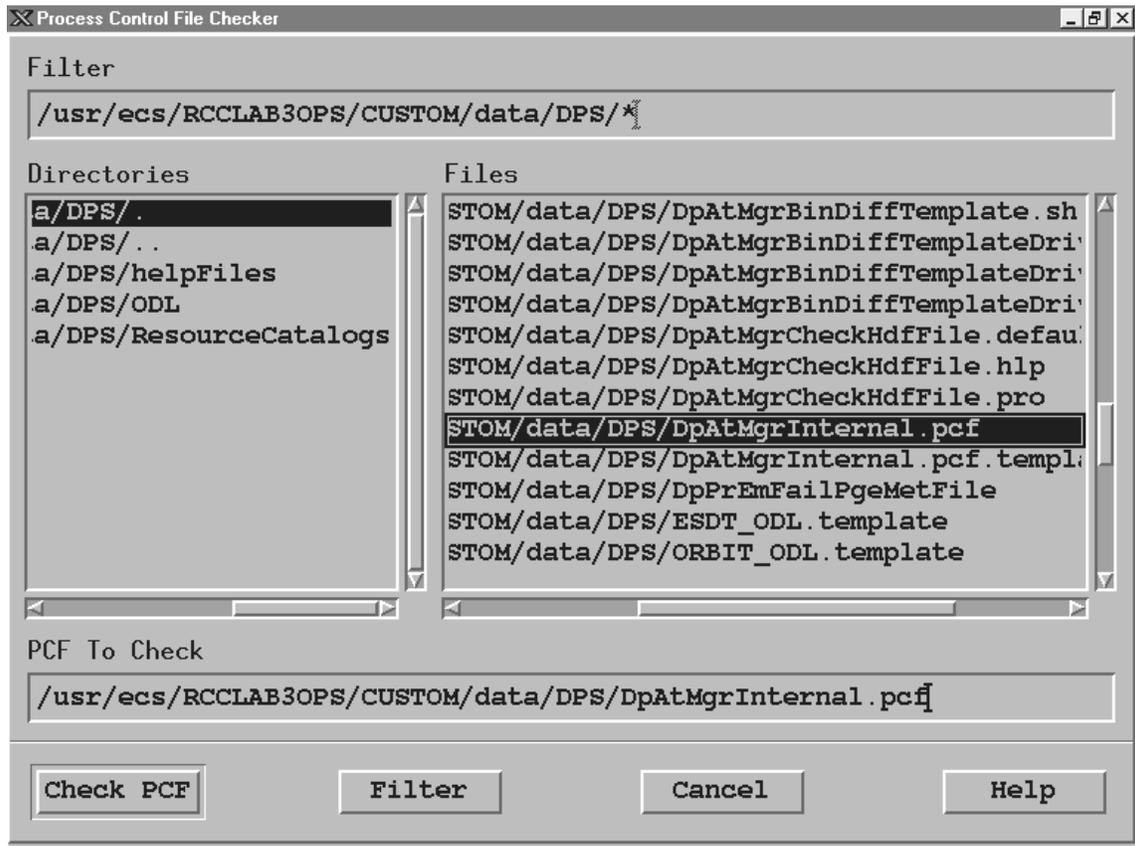


Figure 4.5.1-14. PCF Checker Pop-up

Table 4.5.1-7 describes the PCF Checker's fields.

Table 4.5.1-7. PCF Checker Field Description

Field Name	Data Type	Size	Entry	Description
Filter	ASCII characters	N/A	Read from configuration file/ user input (optional).	Displays the selected directory (from the Directories field).
Directories	ASCII characters	N/A	Read from database.	Displays the existing directories in the database.
Files	ASCII characters	N/A	Read from database (local disk).	Displays the files existing in the selected directory.
PCF To Check	ASCII characters	N/A	Read from database (local disk).	Displays the selected file to be checked.

The PCF Pop-up allows the user to work through the directory structure on the local machine in order to select PCFs to be checked, by selecting the desired directory in the Directories window. The Filter window allows users to limit the files that are displayed. From the File window, the user can select the PCF file to be checked. Only one PCF can be checked at a time. Selecting the Check PCF button initiates the checking process on the PCF specified in the PCF To Check window. The result is displayed in a Pop-up labeled PCF Checker Results, Figure 4.5.1-15.

If the selected file does not exist, an error dialog is displayed to inform the user. Similarly, if the specified PCF exists but can not be opened, an error dialog pops up to inform the user. Dialogs also pop up to inform the user of any fatal system errors or errors encountered in calls to toolkit functions.

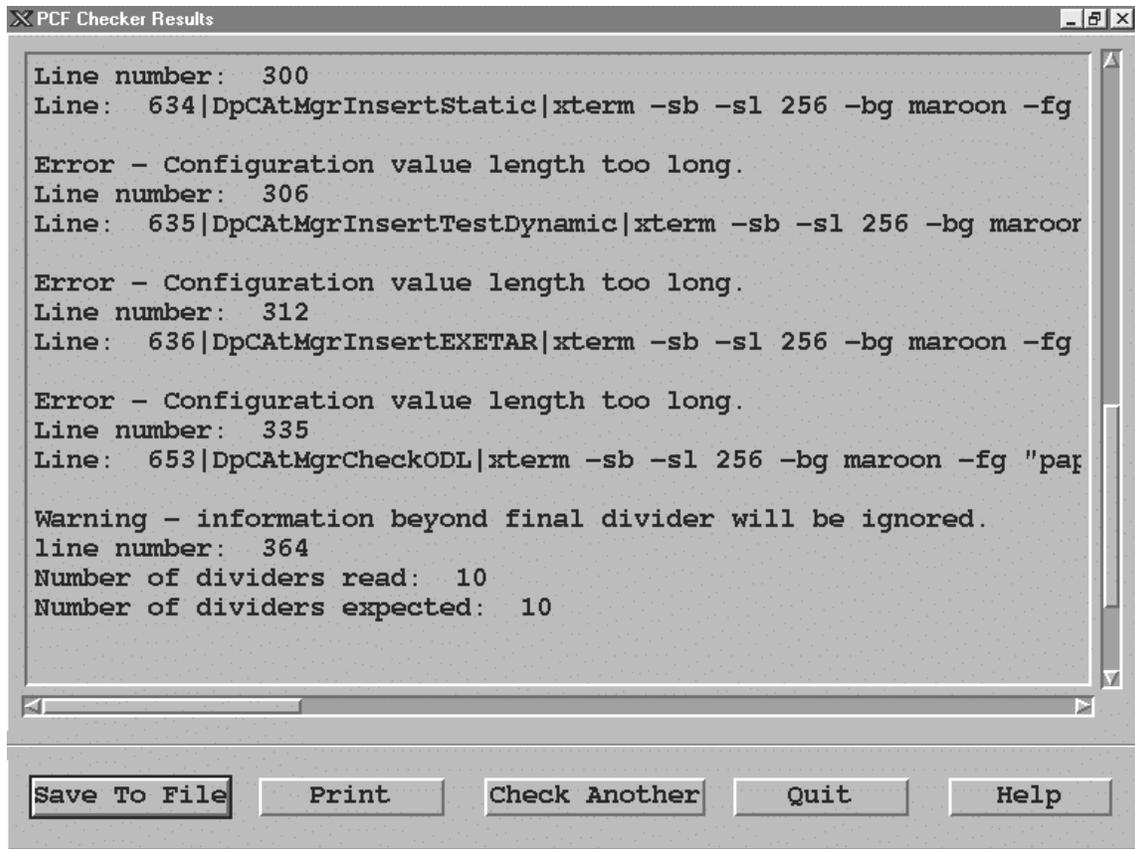


Figure 4.5.1-15. PCF Checker Results Pop-up

From the PCF Check Result Pop-up the user has options to save the results to a file, send the results to the default printer, check another PCF, or quit the Results Pop-up. Selecting the Save button will bring up a Pop-up labeled Save (shown in Figure 4.5.1-16) that allows the user to specify the directory and file name in which to save the results file.

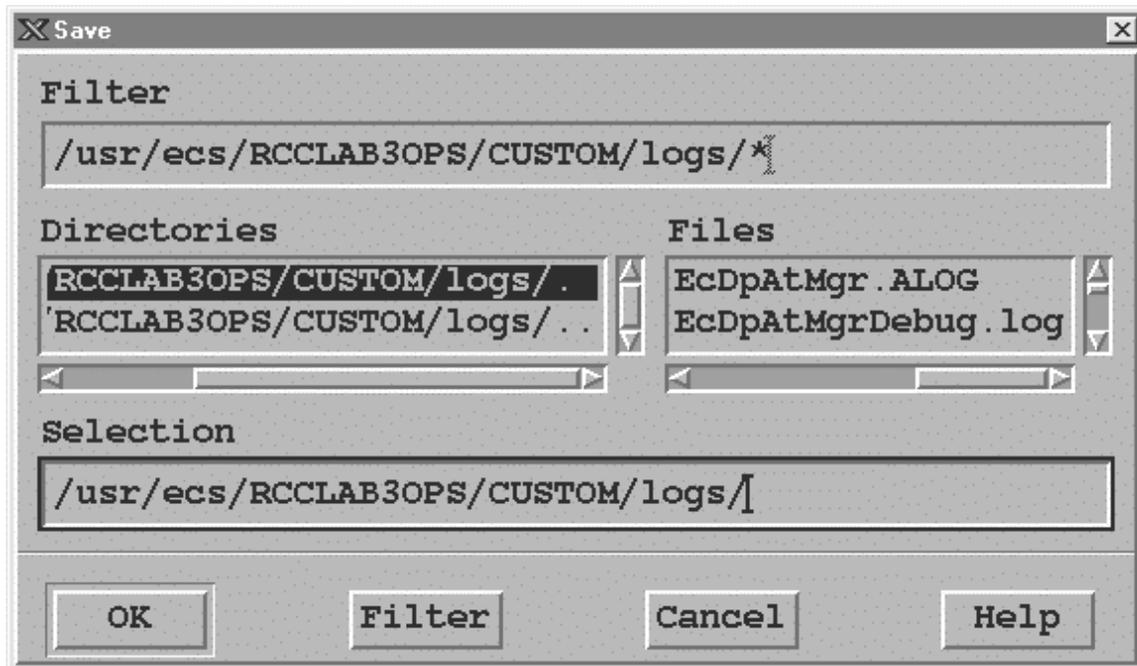


Figure 4.5.1-16. The PCF Checker Save Pop-up.

The PCF Check Save window displays a default output directory and its contents on the file selector. The description fields of the Save Pop-up are identical to the PCF Checker Pop-up's (see table 4.5.1-14). The user may select a listed file or may enter his/her own selection. Selecting the Filter button brings up another screen that allows the user to limit the number of files listed. Figure 4.5.1-17 shows the filter screen.

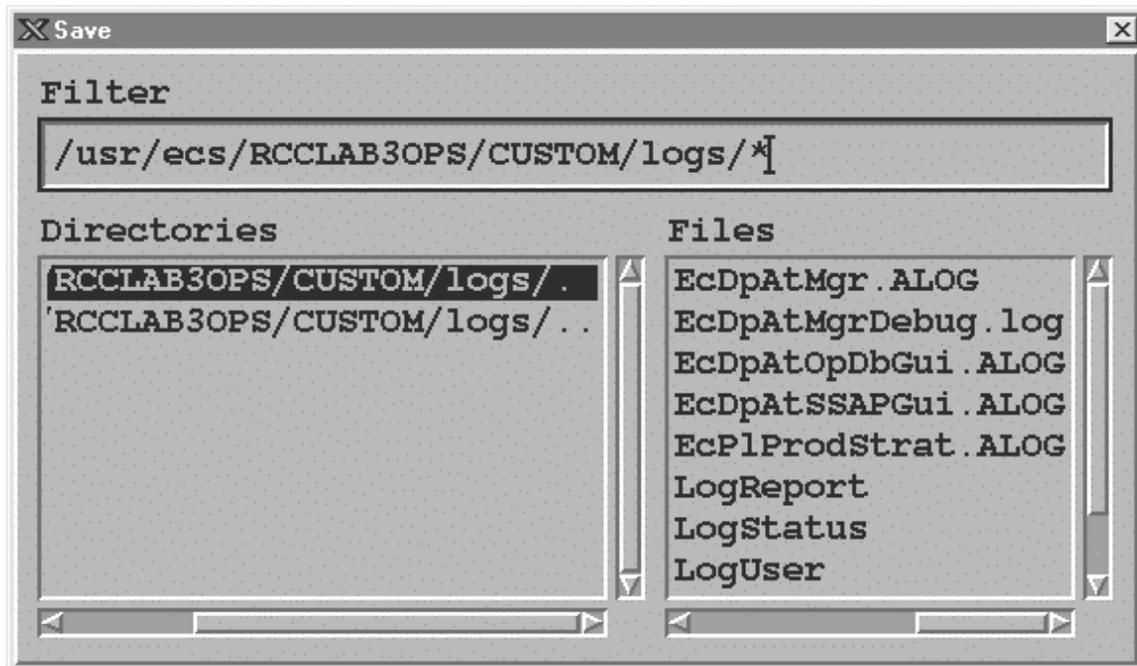


Figure 4.5.1-17. The Pop up display for Filtering files from the Save Pop-up

Messages are displayed to inform the user of successful/unsuccessful completion of save operations.

The Cancel button terminates the running (open) application.

The Help button in all of the PCF Checker Pop-ups provides a description of all the functional buttons for the Pop-up. For example, Figure 4.5.1-18 shows the Help on PCF Checker Results screen.

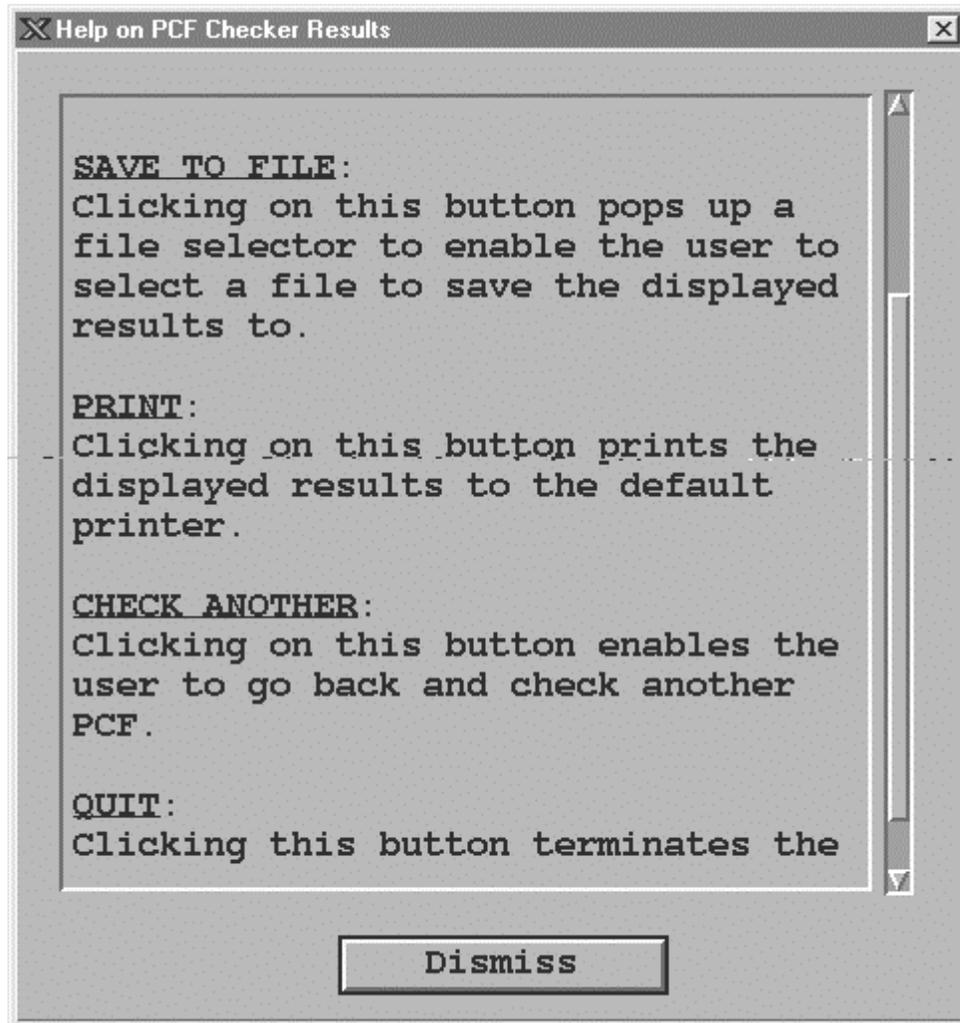


Figure 4.5.1-18. The Help on PCF Checker Results Pop-up.

4.5.1.2.5.4 Prolog Extractor

The Prolog Extractor extracts prologs from science software source code modules.

By default, these prologs are assumed to be surrounded by standard delimiters, according to the official NASA/ESDIS standards document the starting delimiters are list in Table 4.5.1-8.

Table 4.5.1-8. Prolog Extractor standard delimiters

Language	Type	Delimiter
FORTRAN 77	source	!F77
FORTRAN 90	source	!F90
C	sourc	!C
Ada	source	!ADA
FORTRAN 77	include	!F77-INC
FORTRAN 90	include	!F90-INC
C	include	!C-INC
any language	any	!PROLOG

The end delimiter is always !END.

Other delimiters may be specified through use of command line flags (see below).

The Prolog Extractor program will return a warning if the module contains no prologs at all. It does not warn of functions within a particular file that are missing prologs.

In order for this tool to work, the file must have one of the file extensions listed in Table 4.5.1-9.

Table 4.5.1-9. Prolog Extractor file extensions

Language	extension
FORTRAN77:	f, f77, ftn, for, F, F77, FTN, FOR
FORTRAN90:	f90, F90, f, F
FTN. INCLUDE:	inc, INC
C:	c
C include	h
Ada:	a, ada

The Prolog Extractor is invoked from the SSIT Manager under the TOOLS menu item Standards Checkers.

Output from this tool is always written to a file.

By default, the SSIT Manager internal PCF specifies that this output file be named prologs.txt in the local directory. You may change this by editing the internal PCF, item #650.

The PCF value of the output filename is overridden by the "-o" option on the command line.

4.5.1.2.6 Product Examination

Product examination tools are used to view, examine, or compare files used in science processing. Figure 4.5.1-19 shows the Product Examination tool menu options on the SSIT Manager Main screen. These tools assist in the maintenance of the science software by allowing the operator to view programs in different languages, view science data, or compare program or data output files.

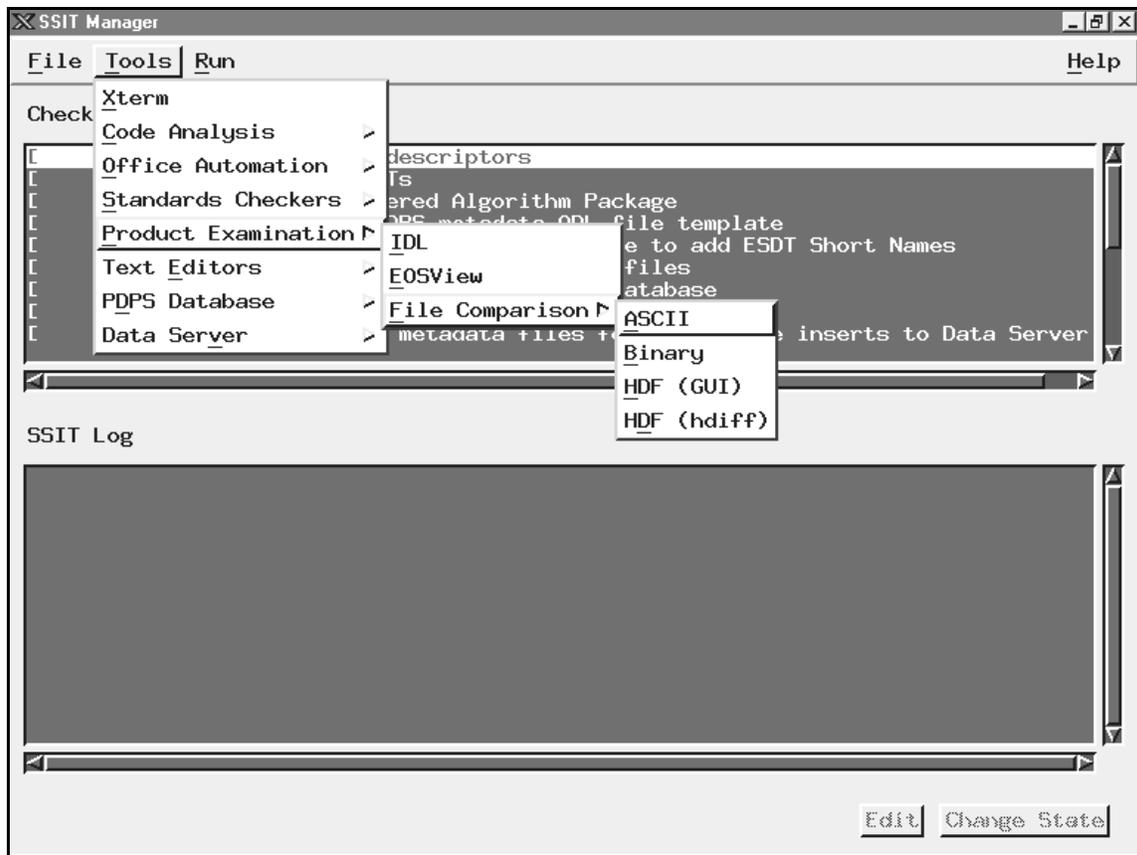


Figure 4.5.1-19. SSIT Manager Main Screen with Product Examination tools displayed

4.5.1.2.6.1 IDL

IDL does not provide a GUI. IDL is a powerful visualization package in common use throughout the science community. This COTS package includes a language, also called IDL, that provides capabilities for data visualization and comparison of various file formats. IDL is provided so that DAAC operators can write data visualization and comparison tools, and execute them on an "as needed" basis.

The IDL language is an interpreted computer language, allowing applications (written in IDL) to look at binary (unformatted) files as well as HDF files.

Once inside this environment, user written programs can be compiled and executed.

Documentation for IDL is obtained by typing `idlhelp` on the UNIX command line.

4.5.1.2.6.2 EOSView

EOSView is described in Section 4.12.5, "EOSView".

4.5.1.2.6.3 File Comparison Tools

This section describes how the File Comparison tools are used by the SSI&T Specialist during the SSI&T process.

The File Comparison tools are selected from the Tools item on the menu bar, and appears under the Product Examination option. Four alternatives are then offered:

- *ASCII* – Text file comparison tool; X windows based xdiff utility
- *Binary* – Custom development environment which assists in the generation of DAAC-written code which compares binary files
- *HDF (GUI)* -- IDL program which compares HDF format files
- *HDF(hdiff)* – “hdiff” CHUI (CHaracter oriented User Interface) utility which compares HDF format files, written by MODIS

These tools would be most frequently used to insure that the data products generated from the PGEs run in the SSI&T environment are identical (within tolerance) to the files that were produced by the PGEs at the SCF.

4.5.1.2.6.3.1 ASCII File Comparison Tool

When the ASCII File Comparison tools is started from the TOOLS/PRODUCT EXAMINATION menu item, an Xterm is spawned, which prompts the user for input. The user enters two ASCII filenames, and the differences between the files is displayed. The Tools is based on the use of the X-Window application xdiff.

4.5.1.2.6.3.2 Binary File Difference Environment

When the Binary File Difference Environment is started from the TOOLS/PRODUCT EXAMINATION menu item, the Pop-up shown in Figure 4.5.1-20 is displayed. The button functions are described below in the order they should be used.

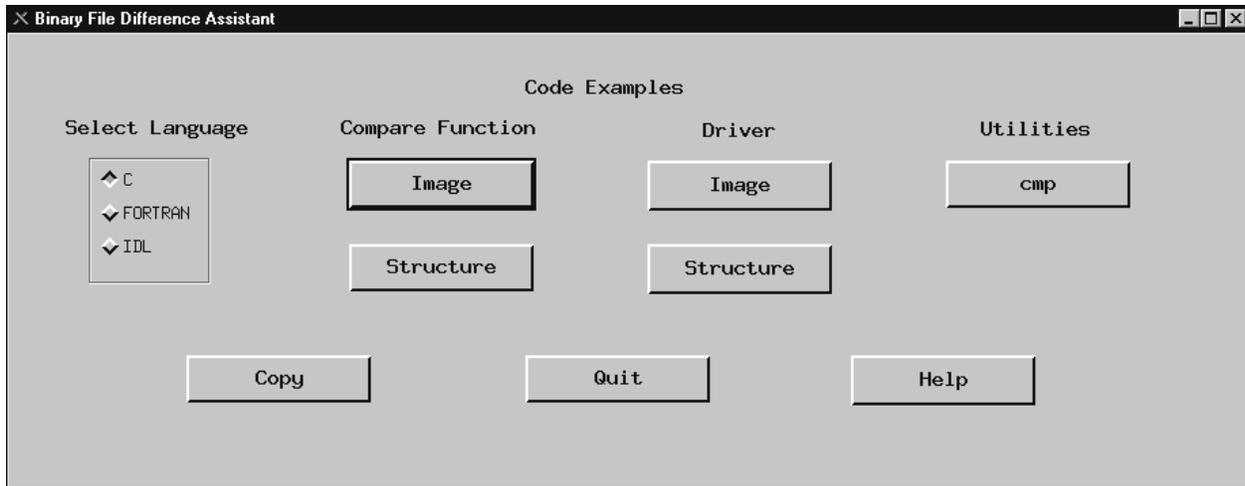


Figure 4.5.1-20. Binary File Comparison Pop-up

The Binary File Difference Assistant assists the user in development of custom code that compares two binary files.

Since there is an unwieldy number of possibilities for binary file formats, this tool cannot compare two binary files without some custom code written at the DAAC, hence, the “Assistant” in the name.

- The **Select Language** box, provides a choice of languages.
- The choice depends largely on preference.
- It does not necessarily have to be the language that was used to create the files being compared. The user clicks “**COPY**” to generate three files: a main driver code module, a template comparison code module, and a makefile in the selected language [C, FORTRAN 77 or IDL (Interactive Data Language)].
- The operator then edits this template to add code that compares files for the specific binary file format, compiles the code, and executes it as appropriate; these operations are all independent of the Pop-up.

The **Code examples** section of the Pop-up displays simple code examples that do binary file comparison. The operator may copy and paste these in their code if desired. Examples have been compiled and tested. This sample code is provided to help the operator in visualizing a completed comparison module. This code is displayed when the user presses one of the “**Compare Function**” buttons, either **Image** or **Structure**. **Image** displays an example of binary_file comparison code for files containing images. **Structure** displays an example of binary file comparison code for files containing arrays of structures, or records.

Also provided is sample driver code, when the operator presses a “**Driver**” button. Finally, code for a simple compare utility is provided for reference, when the user presses “**cmp**”.

4.5.1.2.6.3.3 HDF GUI

All the ECS standard data products are generated in HDF format. A tool is provided to compare HDF data files obtained running some algorithm(s) in a testing versus an operational processing environment. The HDF file difference tool is invoked from the SSIT Manager under the **Tools** menu item **Product Examination**, sub-menu **File Comparison**, sub-menu **HDF (GUI)**.

The tool is also available from the UNIX command line by typing *EcDpAtCheckHdfFile*. Both methods will result in the display of the screen shown in Figure 4.5.1-21 (*note that this program is written in IDL, so it invokes the IDL environment to run.*)

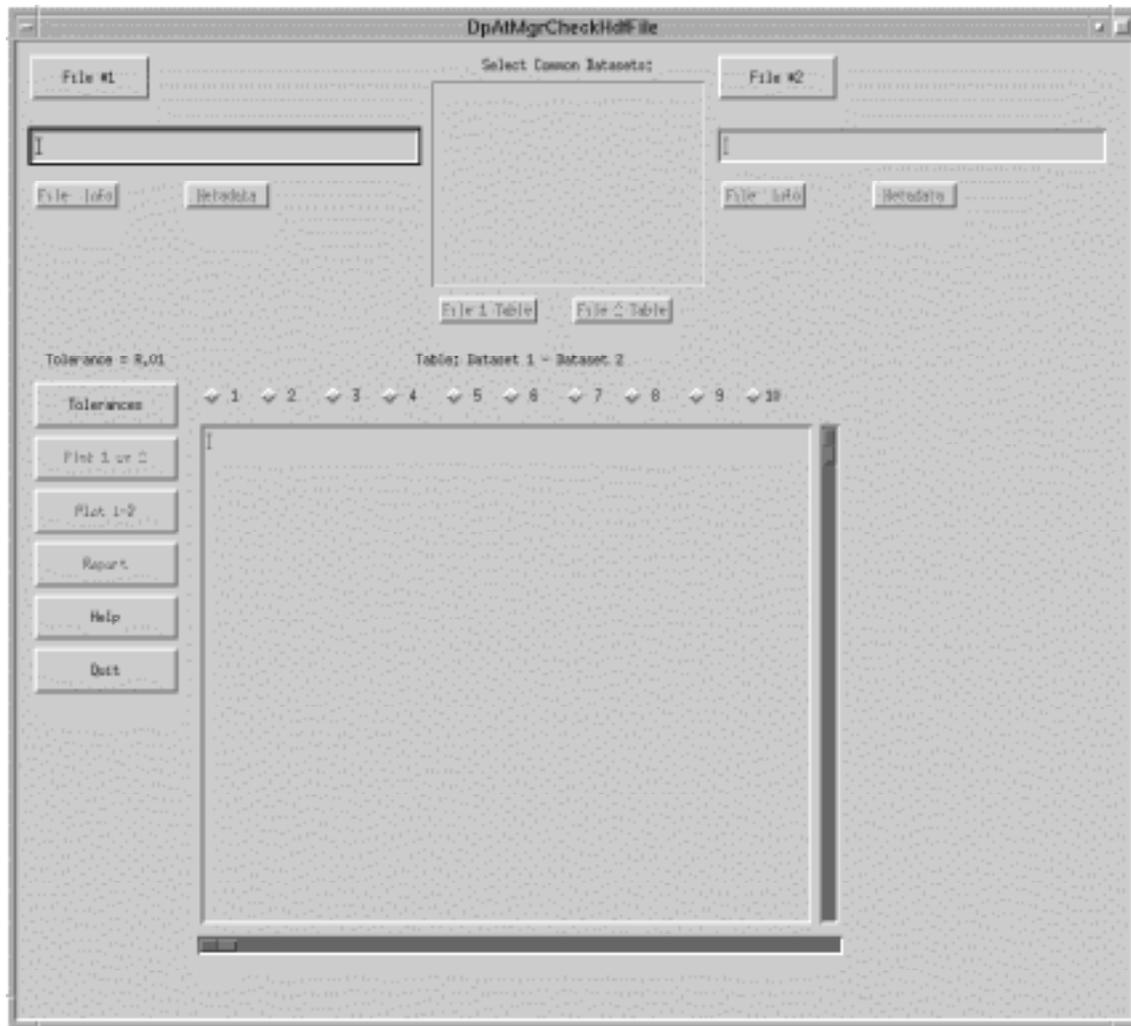


Figure 4.5.1-21. HDF (GUI) Pop-up

The HDF File Difference Tool Pop-up allows the operator to compare and contrast 2 HDF files via a graphic user interface. The button functions on this Pop-up are discussed

- The selection of the files to be compared is done by clicking the **File #1** or **File #2** buttons. This selection activates the File Selection Dialogue Pop-up shown in Figure 4.5.1-22. The appropriate filename is displayed in the Selection textbox of the Dialogue Pop-up. The “OK” button will return to the HDF File Comparison Pop-up and insert the selected filename in the originally activated file selection display textbox.

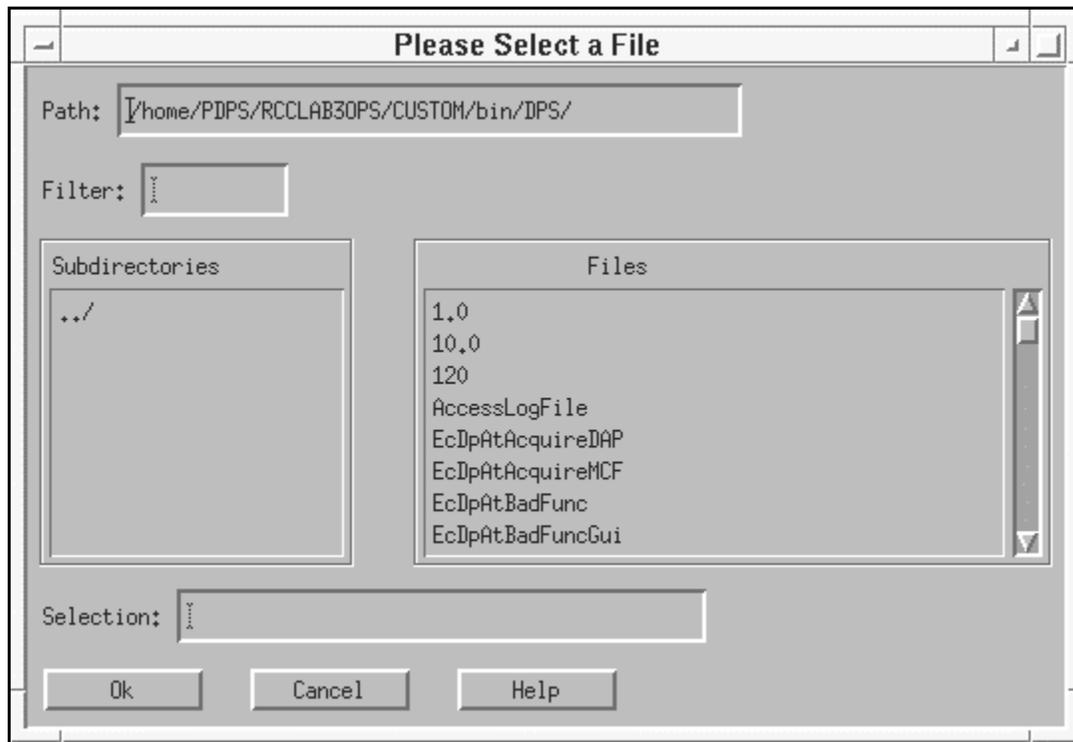


Figure 4.5.1-22. File Selection Dialogue Pop-up

- For each selected HDF file, **File Info** and Associated **Metadata** can be displayed by clicking on the available buttons.
- The Tool provides also the facility to select the **Common Datasets** of both files from the list available in the center upper part of the screen.
- Once the datasets are identified, the operator selects the type of tolerance (relative or absolute) that is used to compare the numerical values of the two datasets. This is accomplished by clicking on the **Tolerances** button (Figure 4.5.1-21) and typing the appropriate value of tolerance as shown in Figure 4.5.1-23.

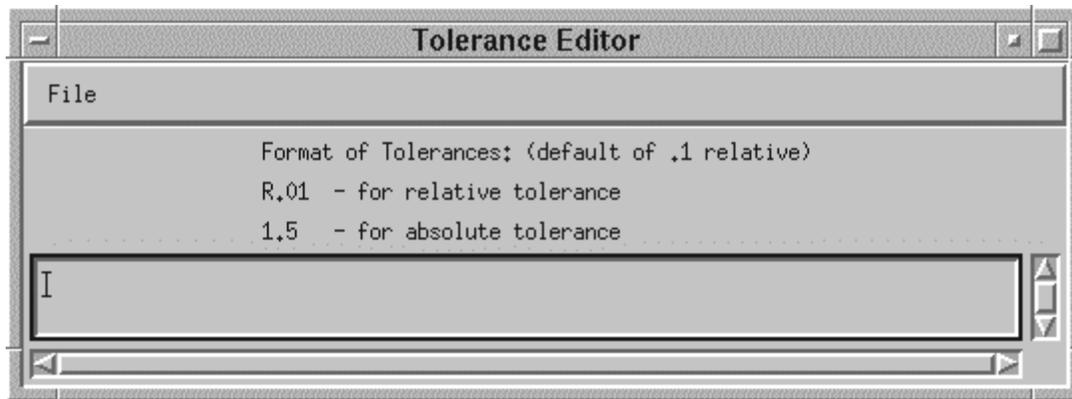


Figure 4.5.1-23. Tolerance Editor Pop-up

- This interface allows the operator to input comparison coefficients (shown in Figure 4.5.1-21 as radio buttons **1** through **10** next to the **Tolerance** button on the HDF File Comparison Pop-up).
- Finally, two types of graphs can be selected by the operator by clicking on either one of the following buttons:
 - **Plot 1 vs 2** shows a plot with both datasets;
 - **Plot 1-2** shows the difference between the two datasets.
- Documentation for the HDF file difference tool is obtained by clicking on the **Help** button on the main screen.
- Clicking on the **Quit** button will close this Tool.

4.5.1.2.6.4 HDF File Comparison - hdiff

The HDF File Comparison hdiff tool spawns a Xterm when it is started from the TOOLS/PRODUCT EXAMINATION menu item which prompts the user for input. There is no graphics screen for this function. It is run through the command line interface. The operator is also provided with a list of options for different kind of comparisons that the tool can perform on HDF files (See Figure 4.5.1-24). After the operator enters two HDF filenames, the differences between the files is displayed.

```
SSIT: HDF File Comparison
Options for HDiff:
[-g]          Compare global attributes only
[-s]          Compare SD local attributes only
[-d]          Compare SD data only
[-D]          Compare Vdata data only
[-v var1[,...]] Compare SD data on variable(s) <var1>,... only
[-u var1[,...]] Compare vdata on variable(s) <var1>,... only
[-e count]    Print difference up to count number for each variable
[-t limit]    Print difference when it is greater than limit

Options for comparison? (enter for null)

```

Figure 4.5.1-24. HDF (hdiff) options

4.5.1.2.7 Text Editors

The Text Editors menu provides tools to assist operators in documentation and administrative tasks (Figure 4.5.1-25).



Figure 4.5.1-25. SSIT Manager Main Screen with Text Editors tool menu displayed.

4.5.1.2.7.1 Emacs

Emacs is a COTS product. Emacs, shown in Figure 4.5.1-26, provides SSI&T users with Editor capabilities. More information about Emacs is available on the Web free from <ftp://prep.ai.mit.edu/pub/gnu/>.

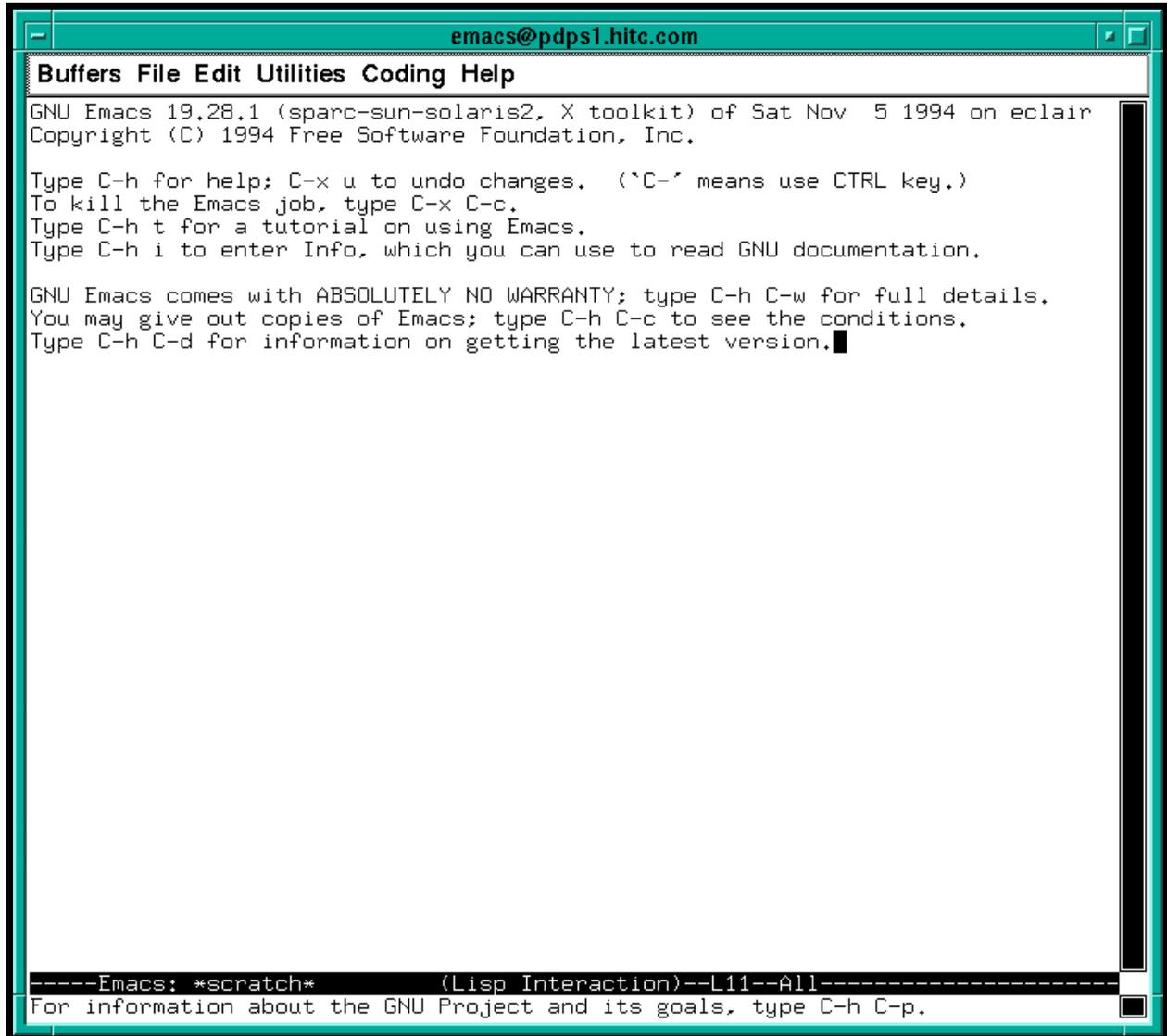


Figure 4.5.1-26. Emacs Pop-up.

4.5.1.2.7.2 Xedit

Xedit is a COTS product. Xedit, shown in Figure 4.5.1-27, provides SSI&T users with Editor capabilities.

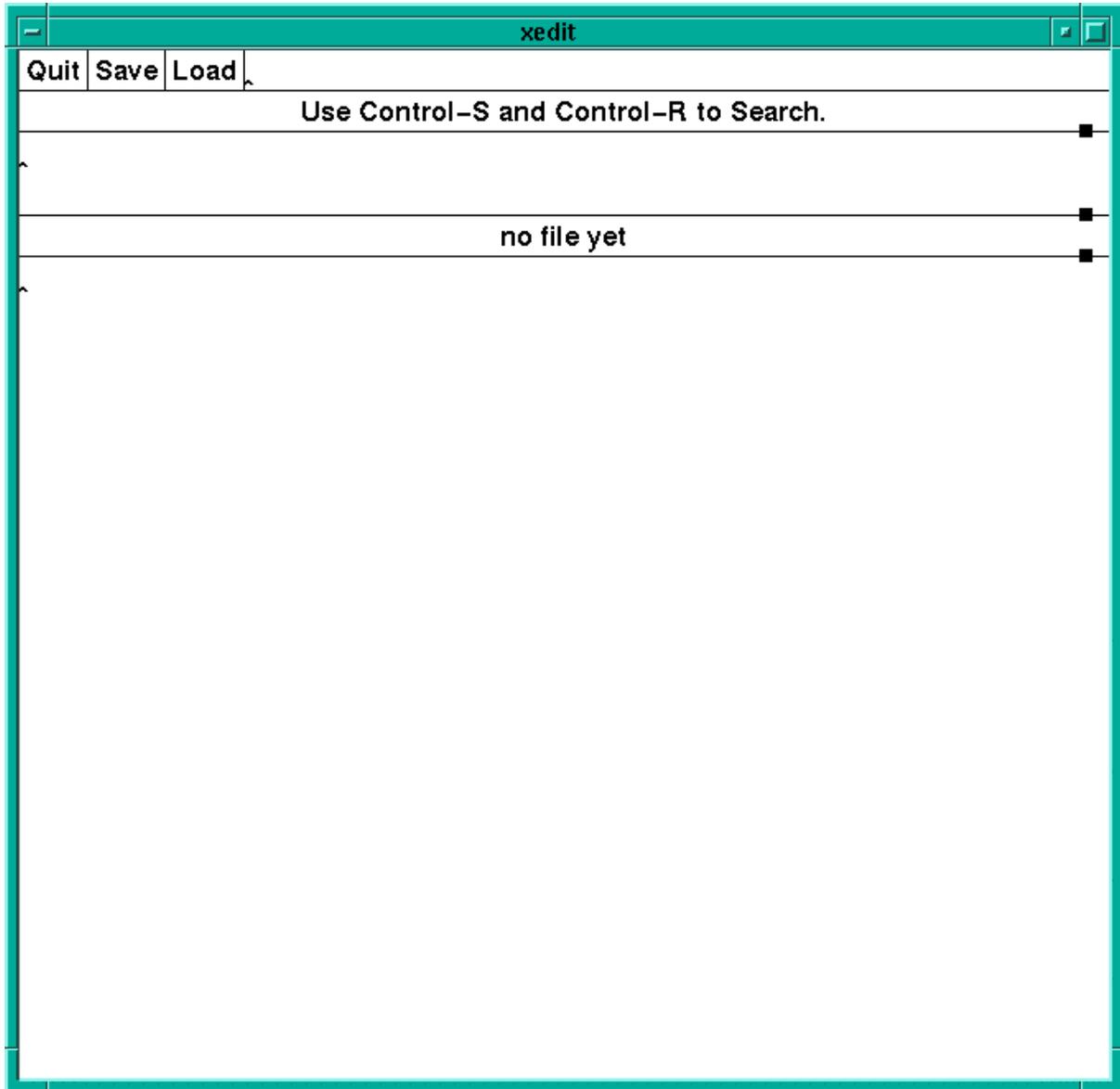


Figure 4.5.1-27. Xedit Pop-up.

4.5.1.2.8 PDPS Database

The PDPS Database Update Tools are accessed from the *TOOLS* menu item of the SSIT Manager Pop-up tool bar, submenu *PDPS Database* as shown in Figure 4.5.1-28. These tools are used to

register a PGE with the PDPS database, during which critical information about the PGE and under what conditions it should be run is captured. This information is used by the PDPS prior to and during PGE execution, in both the test environment and the production environment.

The PDPS Database Update Tools consist of four components;

1. PCF ODL Template
2. Check ODL
3. SSIT Science Metadata Update
4. SSIT Operations Metadata Update (a.k.a. PGE Registration Pop-up).

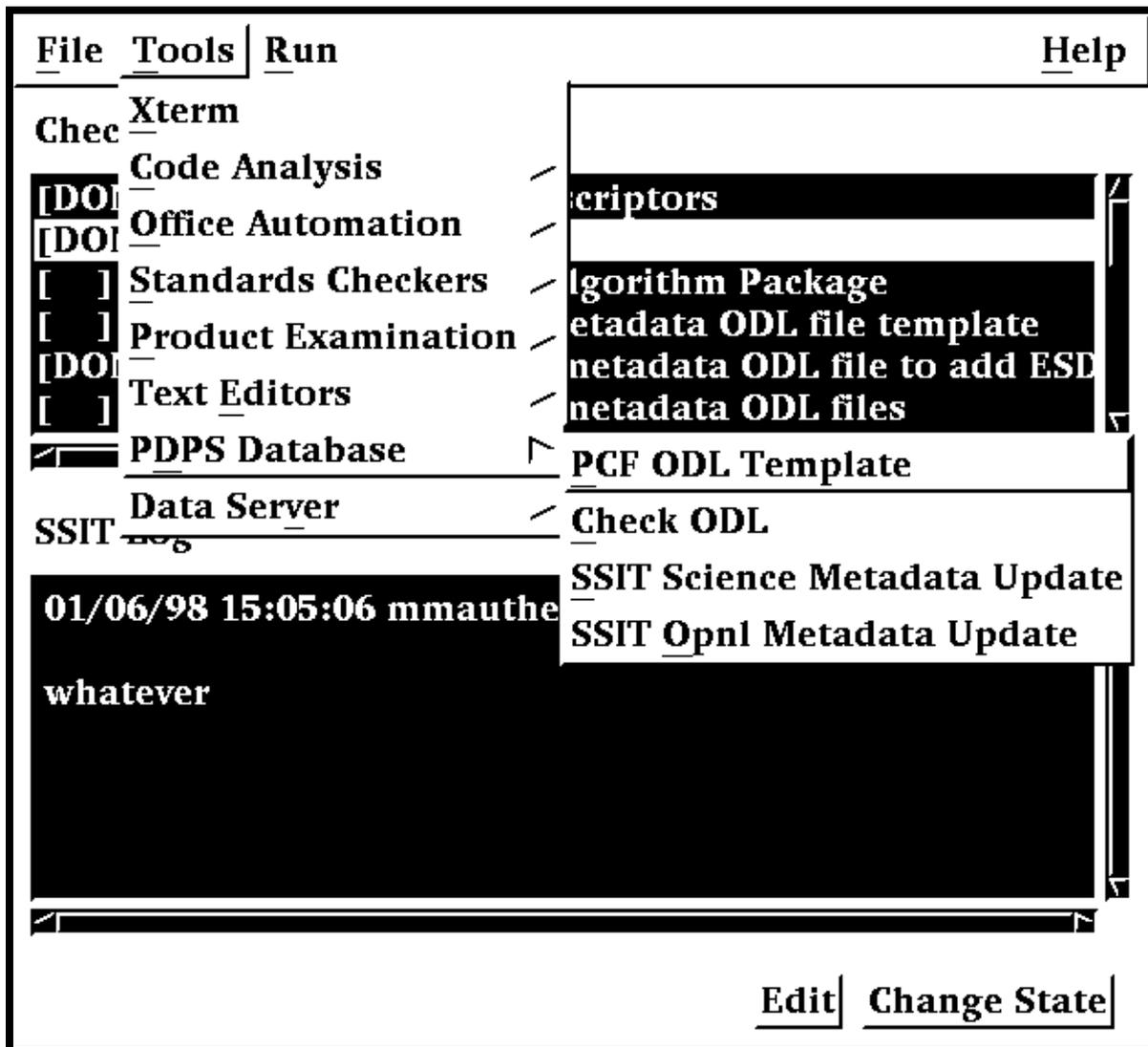


Figure 4.5.1-28. SSIT Manager with PDPS Database tool menu displayed

All programs can be started from the *TOOLS/PDPS Database* menu available on the SSI&T Manager Tool as well as executing the proper command line, as specified in the following sections.

4.5.1.2.8.1 PCF ODL Template and ODL file Checking

The first step in the process of updating the PDPS Database is to generate a template SSI&T PGE SCIENCE metadata ODL file from the PCF delivered with the science software. This is accomplished by selecting *PCF ODL Template* from the *PDPS Database* submenu.

A character based window opens where the user is prompted for various inputs, the configuration filename, the ECS mode of operations, the name of a delivered science software Process Control File (PCF), the PGE name, the PGE version, and the Profile ID.

The tool is also available by sourcing - in the executable directory *\$ECS_HOME/<MODE>/CUSTOM/bin/DPS* - the file named *buildrc*.

After the template file is generated, the SSI&T operator must edit this file, adding all information which is currently blank, as needed. In particular, each PCF file entry must be associated with one or more ESDTs.

The file edited through the previous step can be checked by the Check ODL Script, available through the SSIT Manager via *Tools/PDPS Database/Check ODL*.

The edited file is then used as an input to the next step (see Section 4.5.1.2.8.3) of the PDPS Update process.

4.5.1.2.8.2 Check ODL

The Check ODL program parses the Science Software Metadata ODL files and reports any errors found. It is used to validate ODL (make sure there aren't any errors) before the SSIT Science Metadata Update Tool is executed to populate the Science Software Metadata information in the PDPS database.

The ODL Checker program is command line driven.

The user prompts and explanations follow. Note that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear at the end of the prompt line.

**** SCIENCE Metadata ODL Files Checker Script ****

Configuration filename? (enter for default: ../../cfg/EcDpAtCheckODL.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) that the tool will execute in. In most cases this will be TS1.

PGE name (max 10 characters)?

This is the name of the PGE for which you want to check the ODL. It has a maximum length of 10 characters and must be part of the ODL file name and inside the ODL file. See PGE Metadata ODL File for more information about naming and filling out the PGE ODL file.

PGE version (max 5 characters)?

This is the version of the PGE for which you want to check the ODL. It has a maximum length of 5 characters and must be part of the ODL file name and inside the ODL file. Again, see PGE Metadata ODL File help page for more information about naming and filling out the PGE ODL file.

PGE Profile ID (0-99, 0 means null)?

This is the profile ID of the PGE for which you want to check the ODL. It is an integer with a range 0-99 and must be part of the ODL file name and inside the ODL file. Again, see PGE Metadata ODL File help page for more information about naming and filling out the PGE ODL file.

Any success or error messages will then be displayed followed by:

Hit return to run again, 'q {return}' to quit:

This allows the user to check another set of ODL files (just hit enter) or quit the program (q).

Check ODL program output

The Check ODL program will search for any missing or invalid ODL parameters. It reports errors with the following message:

Check of PDPS Science Metadata was unsuccessful: missing or invalid data in ODL files

What follows is a list of missing or invalid parameters and why they were flagged as invalid. In many cases the parameter value is out of the expected range or is longer than the allowed maximum. For some errors, the Check ODL program cannot narrow the problem to its cause and points the user to the Toolkit Error logs as follows:

DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):

ODL Error in file /ecs/formal/PDPS/DPS/SSIT/src/Metadata/Test/PGE_BAD#ODL#09.odl

Check LogStatus file

User must look in the LogStatus file in the /usr/ecs/{MODE}/CUSTOM/logs directory or in the user-installed directory if the user has installed a personal copy of SSIT.

If the required ODL file(s) cannot be found the tool will report:

DpAtCheckOdlDpAtScienceMd::ProcessOdlFile (PGE):

Unable to open PGE Science Metadata ODL file

{FILENAME}

{FILENAME} will be replaced by the filename that the tool was searching for. If this message appears check the configuration settings (see Preconditions to running the Check ODL program) and that the ODL file exists.

If the check of the ODL succeeds, the user will be greeted with the message:

```
***** Check of PDPS SCIENCE metadata SUCCESSFUL *****
```

```
NO ERRORS FOUND
```

4.5.1.2.8.3 SSIT Science Metadata Update

The next step is to update the PDPS/SSIT database with science metadata, where the latter is defined as PDPS metadata that comes from the Instrument Teams (ITs), and rarely changes. This is done by selecting *SSIT Science Metadata Update* from the *PDPS Database* submenu

The PGE template ODL file, which is output from the *PCF ODL Template* program of the previous section, is input to this program, after it has been edited and renamed.

An example of this file (before editing) is available in:

```
$ECS_HOME/CUSTOM/data/DPS/PGE_ODL.template .
```

The comments in the example file explain fully what data are needed.

The naming convention that should be used for the edited file is explained in the provided Help files. The location of the edited file must be set in the configuration file parameter **DPAT_PGE_SCIENCE_MD**.

An example of this file can be found in:

```
$ECS_HOME/CUSTOM/data/DPS/PGE_ODL.template .
```

In addition, if it does not already exist, a PDPS/SSIT ESDT SCIENCE metadata ODL file must be created, one for each ESDT used by this PGE. Again the naming convention to be used for this file is documented in the Help files. The location of the edited file must be set in the configuration file parameter **DPAT_ESDT_SCIENCE_MD**. An example of this file is provided in:

```
$ECS_HOME/CUSTOM/data/DPS/ESDT_ODL.template .
```

The comments in the example file explain fully what data are needed.

In addition, if it does not already exist, a Production Rule ODL file must be created for this PGE. Again the naming convention to be used for this file is documented in the Help files. The location of the edited file must be set in the configuration file parameter **DPAT_RULE_SCIENCE_MD**. An example of this file is provided in:

```
$ECS_HOME/CUSTOM/data/DPS/ESDT_ODL.template .
```

The comments in the example file explain fully what data are needed.

Once these files (PGE, ESDT, and RULE) have been created and all the blanks have been completely filled in, the user selects *SSIT Science Metadata Update* from the *PDPS Database* submenu to run the program. If a PGE already exists the operator will be prompted to update it. If a PGE does not already exist a character based window opens that prompts the user for information needed to identify the PGE, configuration filename, ECS mode of operations, PGE

name, version and profile ID. With this information the program updates the SSI&T version of the PDPS database with PGE and ESDT Science metadata read from the ODL files.

4.5.1.2.8.4 SSIT Operational Metadata Update

The SSIT Database can be updated with PGE operational metadata.

This Pop-up is used to view/update operational parameters for a particular PGE. Operational Parameters include the performance and the resource requirements for each PGE. In addition users can also view/update PGE user-defined runtime parameter descriptions. Other options include viewing the PGE science metadata file. (The program looks in the configuration file for the **DPAT_PGE_SCIENCE_MD** and **DPAT_ESDT_SCIENCE_MD** files.)

Select View

The initial screen is as depicted in Figure 4.5.1-29, with the *SELECT* tab option displayed. This Pop-up provides the following options.

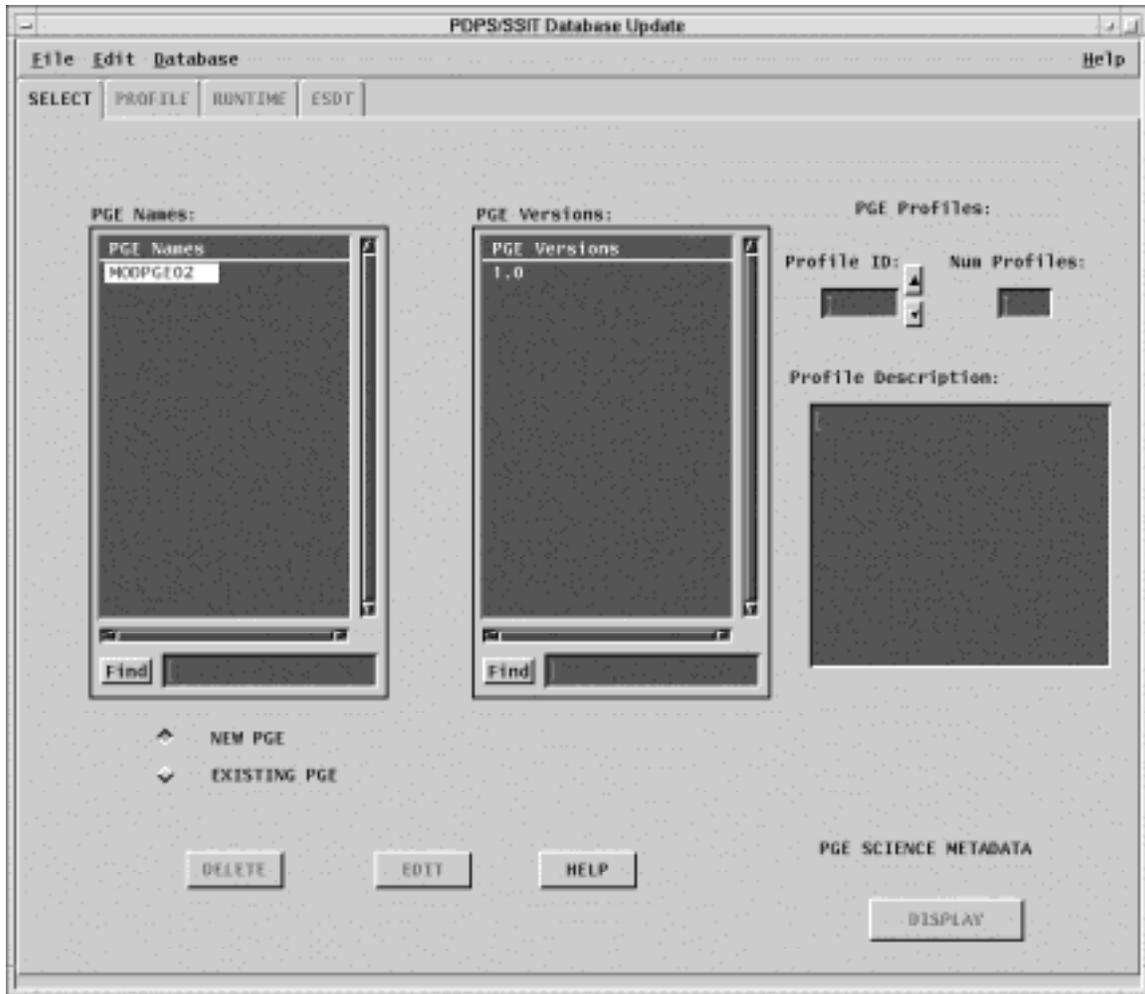


Figure 4.5.1-29. PDPS/SSIT Database Operational Metadata Update Pop-up – showing the SELECT Tab

- The user may select to list either *New* or *Existing PGEs*, the difference being that new PGEs have not yet had their Operational metadata placed into the database.
- Selecting a PGE from the list populates the version list with the all the versions of the selected PGE presently defined in the PDPS database.
- Once the PGE has been selected, information on the PGE profile is also made available to the user through a window that shows a *Profile Description* for each selected *Profile ID*.
- Clicking on the *Display* button displays PGE science metadata in a *Metadata Display* window as read-only.

- If required, selected PGEs can also be flagged as deleted in the database by clicking on the *Delete* button. (However, data is never physically deleted from the database by any SSI&T program. Only database administrators have permission to do this, using ISQL.)

Profile View

Selecting the *Profile* tab displays the profile information of the selected PGE. This view is shown in Figure 4.5.1-30. The view contains fields showing Resource Requirements and Performance Statistics of the selected PGE. Descriptions of each field are presented in Table 4.5.1-10. Default values for these field (i.e. or new PGEs) are set to null. All of the values must be filled in before the database update is completed. For existing PGEs, current settings in the database are displayed. The user can update them as needed. *Cut* and *Paste* capabilities are provided in the *Edit* pull down menu to help performing functionality associated with this tool. Clicking on *Reset* sets the fields to the current values in the database ignoring all the changes operated by the user during the current session. To record updated or new values in the database te user must click on the *Apply* button.

Once the data on this screen has been entered into the database successfully, the PGE changes state from “New” to “Existing”. (The PGE name and version scrolling lists on the SELECT tab are now visible after the user selects *Existing* there.)

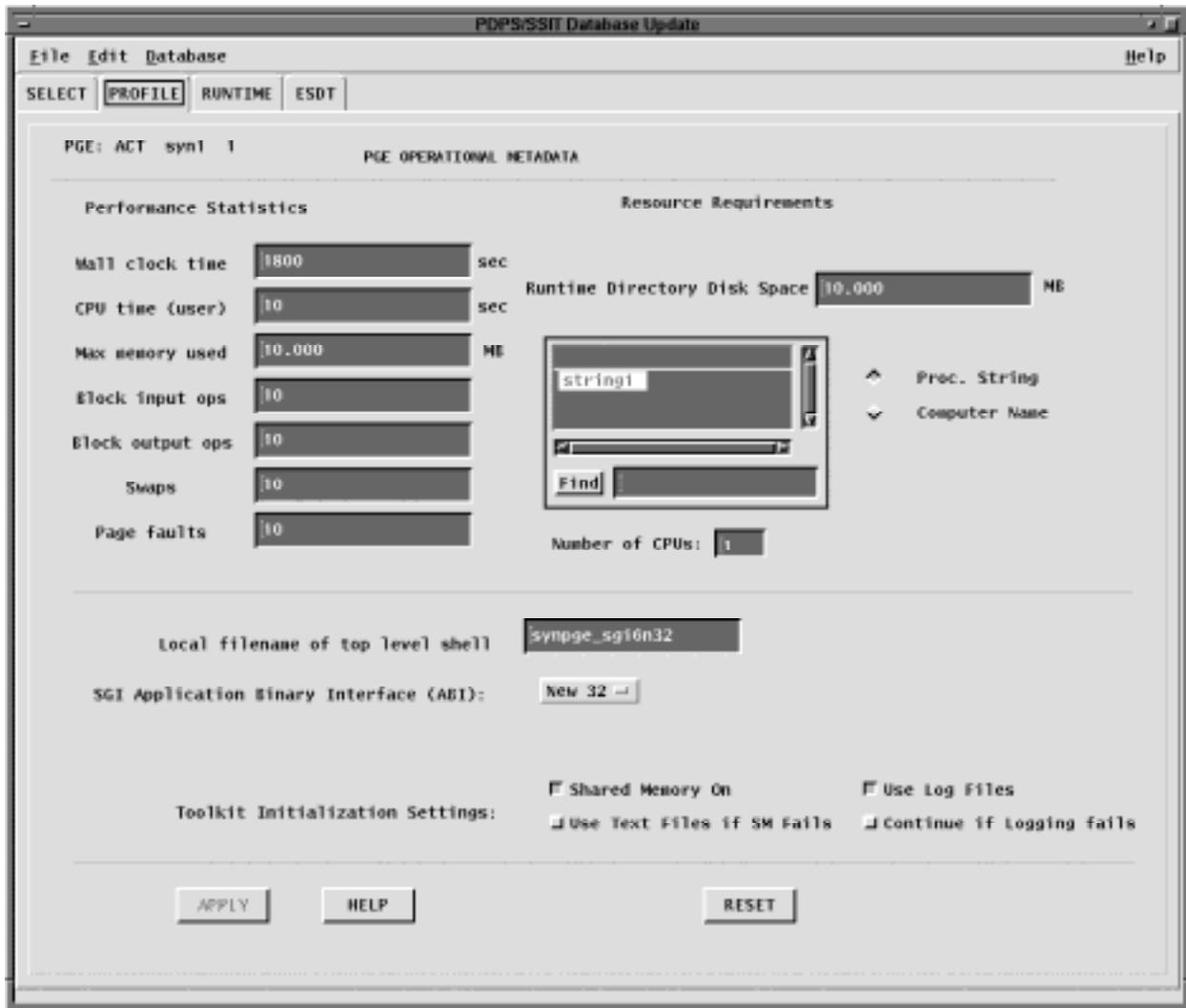


Figure 4.5.1-30. PDPS/SSIT Database Operational Metadata Update Pop-up -showing the PROFILE Tab

**Table 4.5.1-10. PDPS/SSIT Database Operational Update Profile
Field Descriptions (1 of 2)**

Field Name	Data Type	Size	Entry	Description
Wall clock Time	float	N/A	User	Real time used by the process (in seconds); filled in both after processing with SCF Toolkit and with DAAC Toolkit (i.e. with Autosys).
CPU Time	float	N/A	User	CPU time used by the process in seconds.
Max Memory Used	float	N/A	User	Maximum memory used by the process in megabytes.
Block Input Ops	integer	N/A	User	Number of times the file system performed input in servicing.
Block Output Ops	integer	N/A	User	Number of times the file system performed output in servicing.
Swaps	integer	N/A	User	The number of times the process was swapped out of main memory
Page Faults	integer	N/A	User	The number of page faults serviced by the process.
Disk Space used in PGE run	float	N/A	User	Maximum disk space used by the process.
Processing String/Computer Name display Window	Selectable	N/A	system generated	Processing string or computer name on which the PGE will be executed.
Processing String	ASCII characters	30	User	A name given to collection of processors where PGE can be executed.
Computer Name	ASCII characters	30	User	Name of the computer where the process is executed.
Number of CPUs	integer	N/A	User	Number of processors required by the PGE.
Local filename of top level shell	ASCII	30	User	Name of the executable file for the PGE.
SGI Application Binary Interface	Selectable	N/A	System Defined	The "mode" of SGI compilation of the executable.
Toolkit Initialization Settings: Shared Memory On	On/Off	N/A	User	Determines if Toolkit shared memory will be activated during PGE execution.

**Table 4.5.1-10. PDPS/SSIT Database Operational Update Profile
Field Descriptions (2 of 2)**

Field Name	Data Type	Size (of characters)	Entry	Description
Toolkit Initialization Settings: Use Text Files if SM fails	On/Off	N/A	User	Determines if the Toolkit should use text files for shared memory if there is a failure using shared memory.
Toolkit Initialization Settings: Use Log Files	On/Off	N/A	User	Determines if the Toolkit will log information during PGE execution.
Toolkit Initialization Settings: Continue If Logging fails			User	Determines if PGE execution should continue if logging fails.

Runtime View

Selecting the *Runtime* tab displays the PGE runtime parameters, i.e., static parameters used by the PGE at runtime. This view is shown in Figure 4.5.1-31. The logical ID, name and value which have already been entered in the database by the Science Metadata Update program are displayed on the left side of the screen. After selecting one of these runtime parameters the user is given the opportunity to add some descriptive text. (Use of this functionality is optional. Runtime parameter descriptions may remain as null values.) The user clicks the *OK* button when finished editing a particular parameter. Modifications are applied to the database by clicking on the *Apply* button. As before, clicking on *Reset* sets descriptions of all parameters to current values in the database. Details of the runtime parameters fields are shown in Table 4.5.1-11.

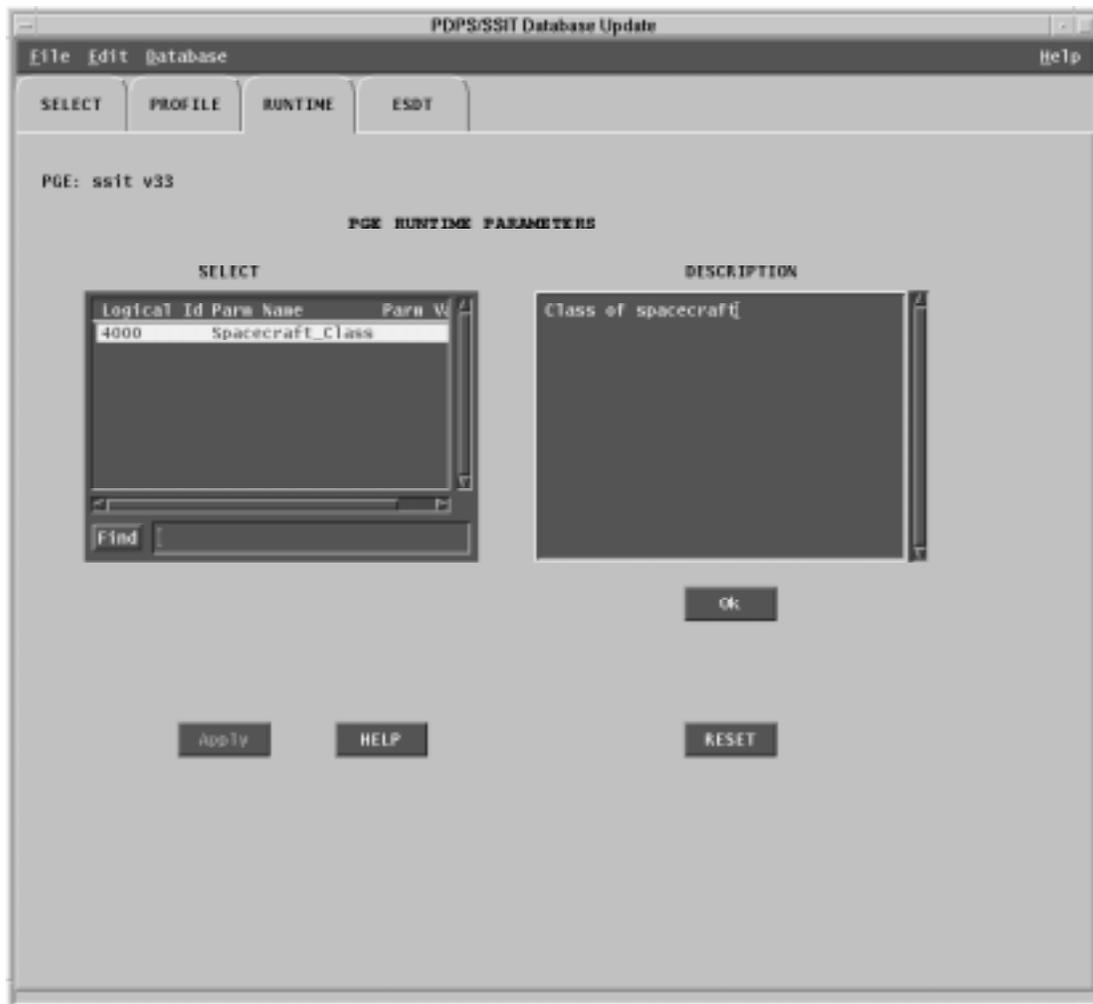


Figure 4.5.1-31. PDPS/SSIT Database Operational Metadata Update Pop-up -showing the Runtime Tab

Table 4.5.1-11 describes the fields on the Runtime Tab.

Table 4.5.1-11. PDPS/SSIT Database Operational Metadata Update Runtime View Field Description

Field Name	Data Type	Size (of characters)	Entry	Description
Logical Id	Integer	N/A	Read from database	logical id of the user parameter in the PCF file
Parameter Name	ASCII characters	N/A	Read from database	User Parameter Name
Parameter Value	ASCII characters	N/A	Read from database	User Parameter Value
User Parameter text description	ASCII characters	<60	User	text describing the user parameter

ESDT View

Selecting the *ESDT* tab displays the ESDT information of the selected PGE. This view is shown in Figure 4.5.1-32. Two ESDTs lists are made available to show both the PGE Input Datatypes and the PGE Output Datatypes. On selection of a particular ESDT, the associated PDPS Science ESDT metadata can be viewed by clicking on the *Display* button. This data is read only and may not be modified through this Pop-up.

The program looks for the file it set in the configuration file parameter **DPAT_ESDT_SCIENCE_MD**.



Figure 4.5.1-32. PDPS/SSIT Database Operational Metadata Update Pop-up - showing the ESDT Tab

Display Screen

Figure 4.5.1-33 shows a pop-up Display screen. The Display window is used to display either a PGE Science Metadata (selected from the *SELECT* tab) or the ESDT Science Metadata (selected from the *ESDT* tab).

A Display screen pops-up also if the request for modification to the database fails because of some mandatory field not being properly set (see screen of *PROFILE* view). Editing is not allowed in this window. Clicking on "Done" closes this window.

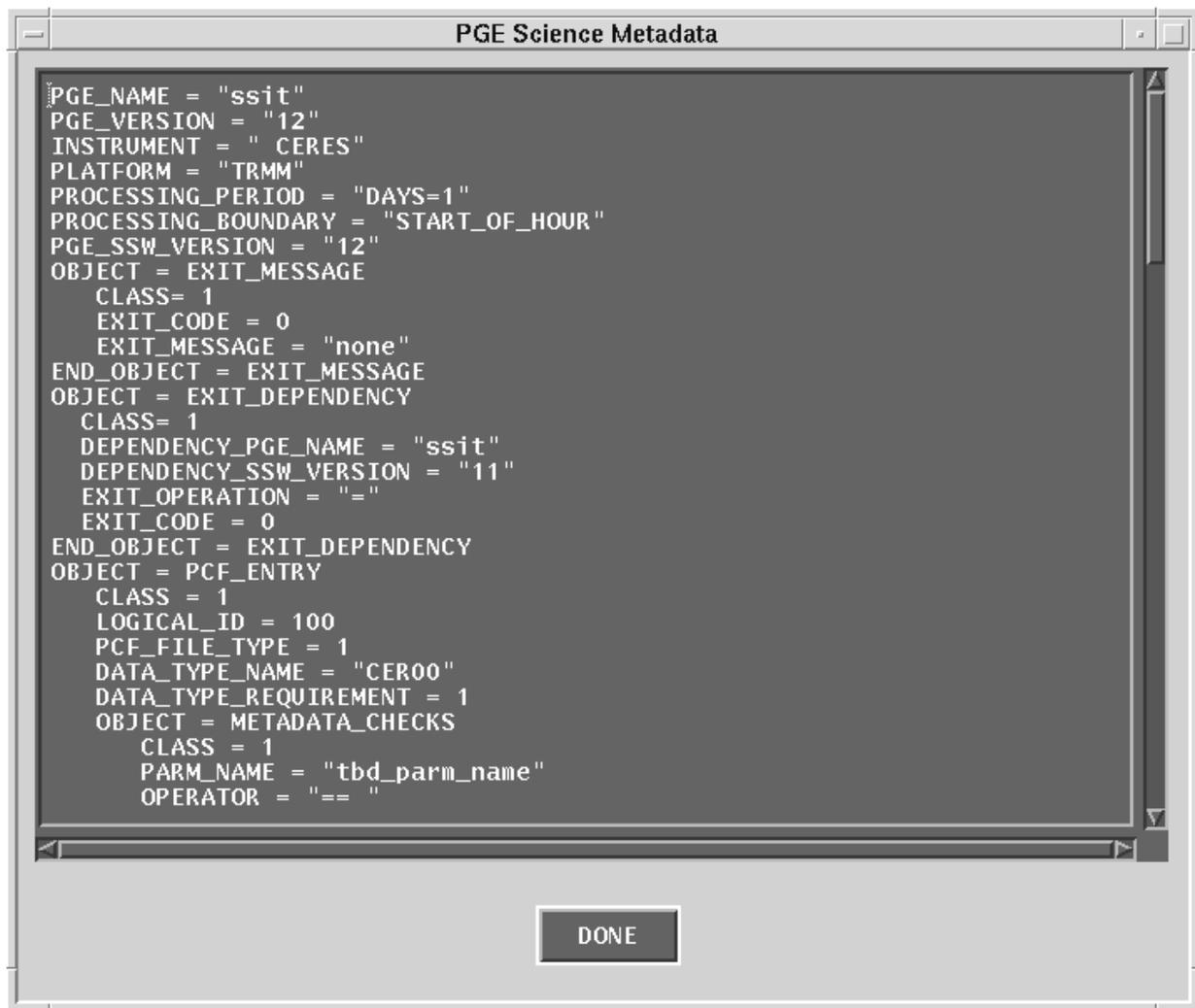


Figure 4.5.1-33. PDPS/SSIT Database Operational Metadata Update Pop-up - Display Screen

4.5.1.2.9 Data Server

This section describes the interfaces to the Data Server used by the SSI&T Specialist during the SSI&T process. All programs may be started from the TOOLS:Data Server menu. Figure 4.5.1-34 shows the submenu choices.

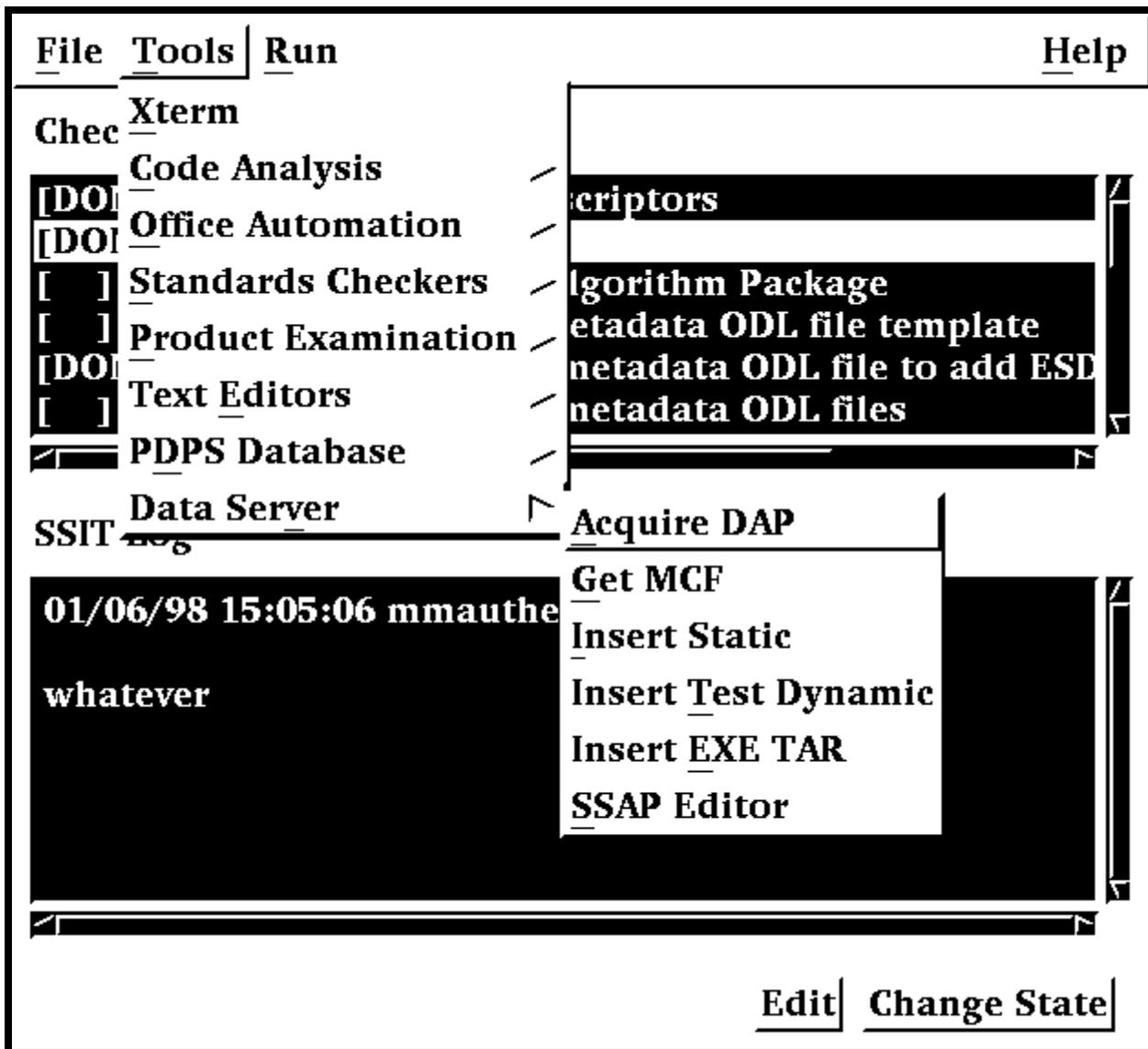


Figure 4.5.1-34. SSIT Manager, Tools Menu, Data Server Submenu Choices

4.5.1.2.9.1 Acquire DAP

This program is used to retrieve the Delivered Algorithm Package (DAP) from the Data Server.

Before its use, the following events must have occurred:

1. A subscription for the DAP must have been registered with the Data Server. The subscription delivery option will be set to email, to a specified DAAC operator or mail drop.

2. The DAP must have been Ingested. There are two ways for this to occur. First, the DAP may be processed by Ingest. When this occurs, Ingest inserts the DAP into the Data Server, triggering subscription notification. Second, the DAP may be inserted with the Insert Test File.
3. After the DAAC operator received the email subscription notification, he or she must have saved it to a file.

To run the program, select *Acquire DAP* from the Data Server submenu. The program prompts for input parameters Process Framework configuration filename, email message filename and directory to receive staged file.

The program reads the DAP UR from the email file, acquires the DAP from the Data Server, and stages the DAP on the local disk. The name of the staged file is written to standard output. Note, this function may be used to acquire any type of granule with a known UR.

After the file is staged, the operator may unpack it and install its components in ClearCase or in other places as appropriate.

NOTE: Use of this program is optional, in the sense that the DAP may arrive at the DAAC through other means than Ingest, e.g., simple ftp.

4.5.1.2.9.2 Get MCF

This tool has a character based user interface. When the tool is selected from the Tools\Data Server\Get MCF menu, the user is prompted to provide pertinent information about the ESDT for the Metadata Configuration File that has to be acquired from the Data Server, including the configuration file name, the ECS mode of operations, the ESDT short name, the ESDT version, and, finally the location where the MCF needs to be transferred to. When the above information is provided the tools contacts the Data Server and, if the operation is successful, copies the MCF to the new location. A message to the operator either confirms the successful transfer of the MCF or issues a warning which explains why the operation did not complete successfully.

4.5.1.2.9.3 Insert Static

The Insert Static File program inserts static input file(s) to the Data Server, for use both during SSIT and in production. Static Files are files which rarely change between instances of PGE processing; for example, calibration files. Such files are normally part of the DAP.

After this program runs successfully, the PDPS database has been updated with the Data Server Universal Reference (UR) of the file inserted. This UR is read by the Planning and Processing systems at runtime. so that the file can be acquired and used during production.

The Insert Test Dynamic File program is command line oriented. There is no CHUI or Pop-up for this program.

Before this program is used, the following must have occurred:

- 1) An ESDT for this data type must have been created in the Data Server.

- 2) A PGE that uses the Static File must have been Registered to PDPS via the SSIT Science Metadata Update tool.
- 3) An ASCII Metadata File (.met) for this ESDT must have been created, in the format expected by the Data Server for Insert.
- 4) Both the static data file(s) in question and its corresponding ASCII Metadata file must be accessible to the local machine.
- 5) The directory where the static data file(s) and the ASCII Metadata File exist must be cross mounted to the Data Server machines.

The Static Input File program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertStaticFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path.

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) of operations. In most cases this will be TS1

ESDT short name for the file(s) to Insert?

This is the ESDT short name (max 8 characters) for the data file(s).

ESDT Version for the file(s) to insert?

This is the ESDT version (an integer) for the data file(s).

Science Group for Static file (one of {C, L, D, O} followed by a 3 digit number)?

This is the Science Group for this Static File(s). It is defined in the PGE Metadata ODL File for each PGE that uses this static file.

C = Coefficient File(s), L = Lookup Table/file(s), D = Database File(s), O = Other Files.

Note that using the wrong letter for the file(s) (say L for a Coefficient File) will not result in an error.

Is there is more than one data file to this Static (Y = Yes, N = No)?

This indicates to the program if there is more than one file associated with the static input. Most static inputs are a single data file and a corresponding ASCII Metadata File. Other static inputs consist of more than one data file and a single ASCII Metadata file (there is always only 1 ASCII Metadata File).

If there is only one data file enter N (or just hit enter for the default). For multiple data files enter Y.

If it is NOT a Static Multi-File Granule (there is a single data file) the following prompts will need to be answered:

Single Filename to Insert (including FULL path)?

This is the name of the data file to insert to the Data Server. Include the full path to the file so that it can be found by the tool.

Associated ASCII Metadata Filename to Insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the data file. Full path to the file must be included so that it can be found by the tool. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

If it IS a Static Multi-File Granule (more than 1 data file) the following prompts will need to be answered:

Directory where all data files and .met file exist (FULL path)?

This is the directory location where all the data files and the ASCII Metadata file are located. For Multi-File Statics all files and Metadata must reside in the same directory.

Name of MFG file (enter to end list)?

This is the name of one of the data files for a Static Multi-File Granule. Enter a file at the prompt and the prompt will then be repeated, allowing for the next file to be named. When all data files have been entered, just hit the enter key at the next prompt (thus entering nothing).

Associated ASCII Metadata Filename to Insert?

This is the name of the ASCII Metadata File (.met) associated with the data file. No path is needed because it is assumed that the file resides in the directory specified above. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

Any success or error messages will then be displayed followed by:

Hit return to run again, 'q <return>' to quit:

This allows the user to enter another Static File (just hit enter) or quit the program (q).

4.5.1.2.9.4 Insert Test Dynamic

The Insert Test Dynamic File program inserts test dynamic input files to the Data Server, for use both during SSIT and in production. Dynamic Files are files which change at each instance of PGE processing; for example, Level 0 data files. This tool performs a function that is normally done by Ingest. It allows the user to insert dynamic files to the Data Server for testing purposes. Normally, Ingest will take in such files and insert them as part of its normal processing.

After this program runs successfully, the specified file(s) have been stored at the Data Server and can be acquired by PDPS when executing PGEs. Also, any subscriptions made on the ESDT of the insert file(s) will trigger, and could (via Subscription Manager) cause PGE waiting on this data to initiate execution

The Insert Test Dynamic File program is command line oriented. There is no CHUI or Pop-up for this program.

Before this program is used, the following must have occurred:

1. An ESDT for this data type must have been created in the Data Server.
2. A PGE that uses the Dynamic File(s) ESDT must have been Registered to PDPS via the SSIT Science Metadata Update tool.
3. An ASCII Metadata File (.met) for this instance of the ESDT must have been created in the format expected by the Data Server for Insert.
4. Both the test dynamic data file(s) in question and its corresponding ASCII Metadata file must be accessible to the local machine.
5. The directory where the dynamic data file(s) and the ASCII Metadata File exist must be cross mounted to the Data Server machines.

The Insert Test Dynamic program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) of operations. In most cases this will be TS1

ESDT short name for the file(s) to Insert?

This is the ESDT short name (max 8 characters) for the data file(s).

ESDT Version for the file(s) to insert?

This is the ESDT version (an integer) for the data file(s).

Is there is more than one data file to this Dynamic Granule (Y = Yes, N = No)?

This indicates to the program if there is more than one file associated with the dynamic input. Most dynamic inputs are a single data file and a corresponding ASCII Metadata File. Other dynamic inputs consist of more than one data file and a single ASCII Metadata file (there is always only 1 ASCII Metadata File).

If there is only one data file enter N (or just hit enter for the default).

For multiple data files enter Y. If it is NOT a Dynamic Multi-File Granule (there is a single data file) the following prompts will need to be answered:

Single Filename to Insert (including FULL path)?

This is the name of the data file to insert to the Data Server. Include the full path to the file so that it can be found by the tool.

Associated ASCII Metadata Filename to Insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the data file. Full path to the file must be included so that it can be found by the tool. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

If it IS a Dynamic Multi-File Granule (more than 1 data file) the following prompts will need to be answered:

Directory where all data files and .met file exist (FULL path)?

This is the directory location where all the data files and the ASCII Metadata file are located. For Multi-File Dynamics all files and Metadata must reside in the same directory.

Name of MFG file (enter to end list)?

This is the name of one of the data files for a Dynamic Multi-File Granule. Enter a file at the prompt and the prompt will then be repeated, allowing for the next file to be named. When all data files have been entered, just hit the enter key at the next prompt (thus entering nothing).

Associated ASCII Metadata Filename to Insert?

This is the name of the ASCII Metadata File (.met) associated with the data file. No path is needed because it is assumed that the file resides in the directory specified above. It can be created by using the Get MCF tool to get a Metadata Control File which specifies the fields and type of data that is required for the ASCII Metadata file.

Any success or error messages will then be displayed followed by:

Hit return to run again, 'q <return>' to quit:

This allows the user to enter another Dynamic File (just hit enter) or quit the program (q).

4.5.1.2.9.5 Insert EXE TAR

The Insert Exe Tar File program is used to insert the tar file containing the PGE executable (and associated files) to the Data Server for use during SSIT and in production.

The Insert EXE TAR program is command line driven. There is no CHUI or Pop-up for this program.

Preconditions to running the Insert Test Dynamic File program

Before this program is run, the following must have occurred:

- An ESDT for EXE TAR files must have been installed at the Data Server.
- The PGE must have been Registered to PDPS via the SSIT Science Metadata Update tool.
- An ASCII Metadata file (.met) for the PGEEEXE ESDT must have been created, in the format expected by the Data Server for Insert. A template for such a file is delivered with the software and stored under /usr/ecs/{MODE}/CUSTOM/data/DPS/PGEEEXE.met.template.
- The tar file must be prepared with the files needed to execute the PGE (see below for a list).
- Both the tar file in question and its corresponding MCF must reside on the local machine.
- The directory where the tar file and the ASCII Metadata File exist must be cross mounted to the Data Server machines.

The Exe Tar file must include

- ⇒ PGE compiled binary executable(s),
- ⇒ PGE scripts (if any),
- ⇒ SCF-generated SDP Toolkit compiled runtime message files (if any),
- ⇒ Any dynamic link libraries needed by the PGE at runtime.

...in short, anything needed to run the PGE, aside from Data Server input and output files.

The program prompts for input parameters Process Framework configuration filename, database name to update, PGE name, science software version, full path filename to insert, MCF full path filename, and top level shell filename within tar file. (“Top level shell filename“ is the name of the script file or executable which one uses to kick off the PGE.)

After the program is run, the PGE is ready to run either in the SSIT environment or in the production system. Then the PDPS database has been updated with the Data Server Universal Reference (UR) of the Exe Tar file inserted, the size of this file, and the top level shell within the tar file. This data is read by the Processing system at runtime and the PGE Exe Tar file is acquired from the Data Server.

NOTE: All Exe Tar files are inserted to a single ESDT, the Short Name of which is hard-coded into the EcDpAtInsertExeTarFile software as PGEEEXE.

The Insert EXE TAR program produces prompts for the operator to enter the necessary information. These prompts and explanations follow. Note that if there is a default value for an entry (in most cases this will only occur if you run the program more than once) it will appear at the end of the prompt line.

Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)

In most cases hitting enter (for the default) is fine. If not, enter the correct configuration filename including full path

ECS Mode of operations?

This is the mode (i.e. OPS, TS1) of operations. In most cases this will be TS1.

Name of PGE?

This is the name of the PGE to insert (max length 10 characters). It must have already been defined to the PDPS database.

Science Software version of PGE?

This is the Science Software version (version of the actual executable) of the PGE (max 5 characters). In most cases this will be the same as the PGE Version, but it is possible to change the PGE Version without changing the actual executable (in which case the Science Software version and PGE Version would be different). The Science Software version is specified in the PGE Metadata ODL File.

Staged filename to Insert (including FULL path)?

This is the name of the Exe tar file to insert to the Data Server. Include the full path to the file so that it can be found by the tool.

Associated ASCII Metadata Filename to insert (including FULL path)?

This is the name of the ASCII Metadata File (.met) associated with the tar file. Full path to the file must be included so that it can be found by the tool. Use the /usr/ecs/{MODE}/CUSTOM/data/DPS/PGEEXE.met.template file to create one for the tar file to be inserted.

Top level shell filename within tar file?

This is the name of the top-level executable within the tar file. It is the name of the program that PDPS is supposed to execute when running the PGE.

4.5.1.2.9.6 Science Software Archive Packages (SSAP) Editor

The SSAP Editor Pop-up is a tool that manages Science Software Archive Packages. In particular, the SSAP Pop-up allows the operator to create a new SSAP, update or delete an existing one, add or remove components, and edit Metadata associated with an SSAP.

From the SSI&T Manager menu, the operator selects **Tools**, **Data Server**, and **SSAP Editor** to start the tool's main window, shown in Figure 4.5.1-35.

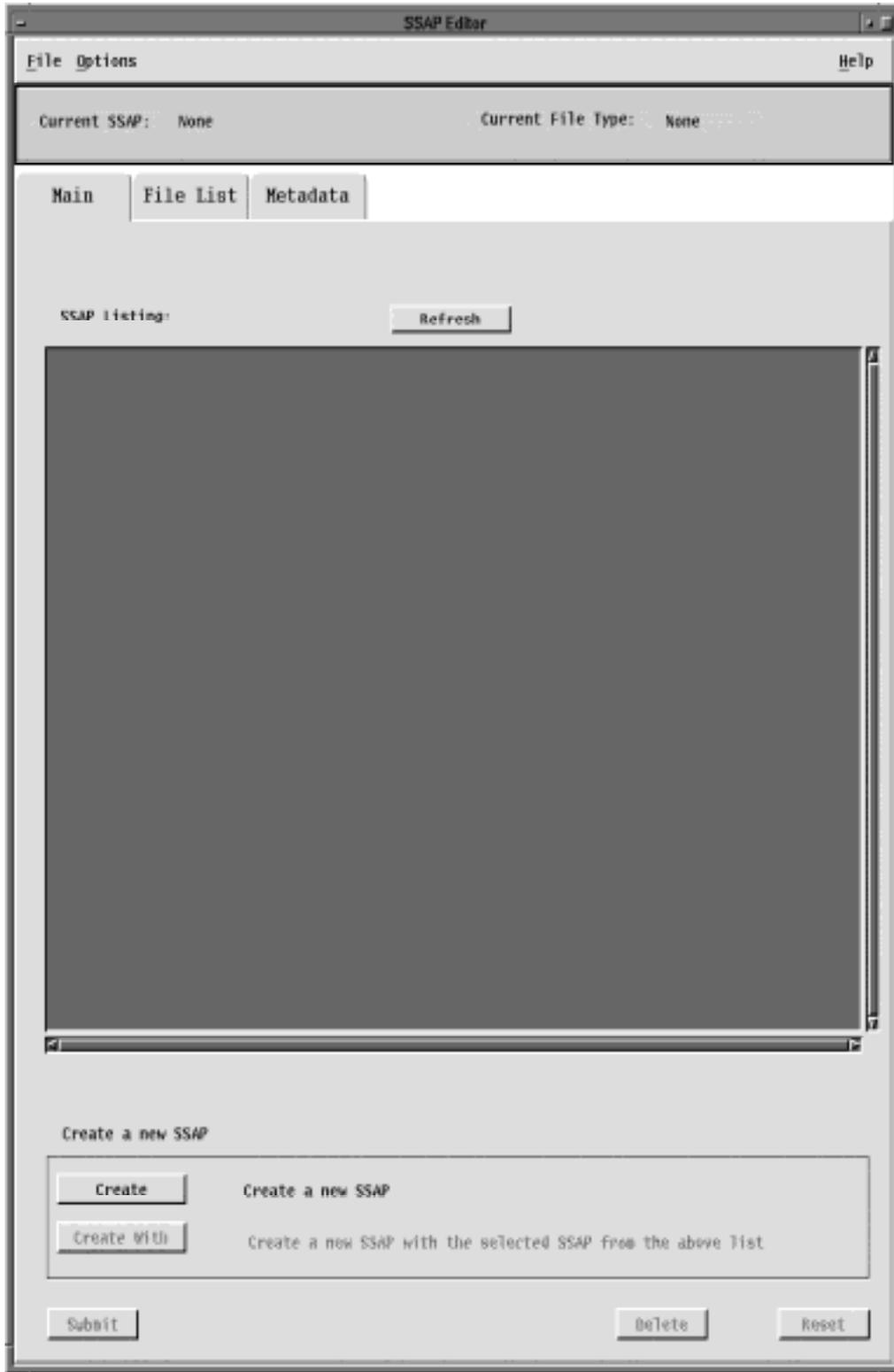


Figure 4.5.1-35. SSAP Editor Main Pop-up

A command line is also available to start the SSAP Editor. The operator can type the command *EcDpAtSSAPGui ConfigFile <CONFIG_FILE> ecs_mode <MODE>* , where <CONFIG_FILE> and <MODE> are, respectively, the name of the Process Framework configuration file for this specific application and the mode of operations. The tool will not allow the operator to come up in a mode that is not consistent with the one established at startup.

The main window contains three pull down menus, each one performing the following functions:

- **The File** menu is used to exit from the SSAP Editor.
- **The Options** menu has two items:
 - **Check Permissions**; displays a Message window that shows a list of functions the operator can perform with the SSAP Editor tool, including; Create SSAPs, Delete SSAPs, Add Files to SSAPs, Delete Files from SSAPs, Edit SSAP Metadata.
 - **Batch Mode**; allows the operator to run a script including command lines for creating SSAPs. The operator is prompted to select the batch mode file using the **Input File Selection** screen, shown in Figure 4.5.1-36.

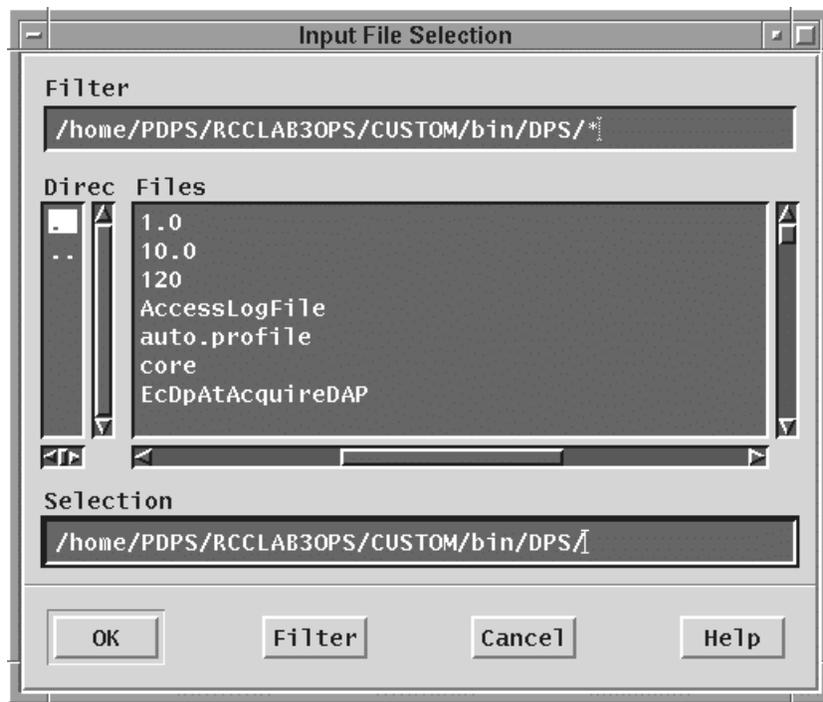


Figure 4.5.1-36. Input File Selection Pop-up

- **The Help** menu provides context sensitive help information.

A toolbar at the top of the screen shows the name of the **Current SSAP** and the **Current File Type** (SSAP), shown in Figure 4.5.1-35.

The SSAP Editor Pop-up main screen has three tabs: **Main**, **File List** and **Metadata**.

The following three subsections provide details on the views that are opened by clicking on each of these tabs.

4.5.1.2.9.6.1 SSAP Editor Main View

The **Main** View (Figure 4.5.1-35) is the default view that appears when the SSAP Editor is started. It contains the following controls:

- A free text window that lists all the existing SSAPs that are retrieved from the Data Server.
- A **Refresh** button that updates the above SSAP Listing, regenerating the screen every time the operator needs to do so, to add recent changes to the listing.
- A **Create** button that allows the operator to start the creation of a new SSAP. Upon clicking this button, the New SSAP screen appears where the operator is prompted to provide a new SSAP Name and SSAP Version, shown in Figure 4.5.1-37.

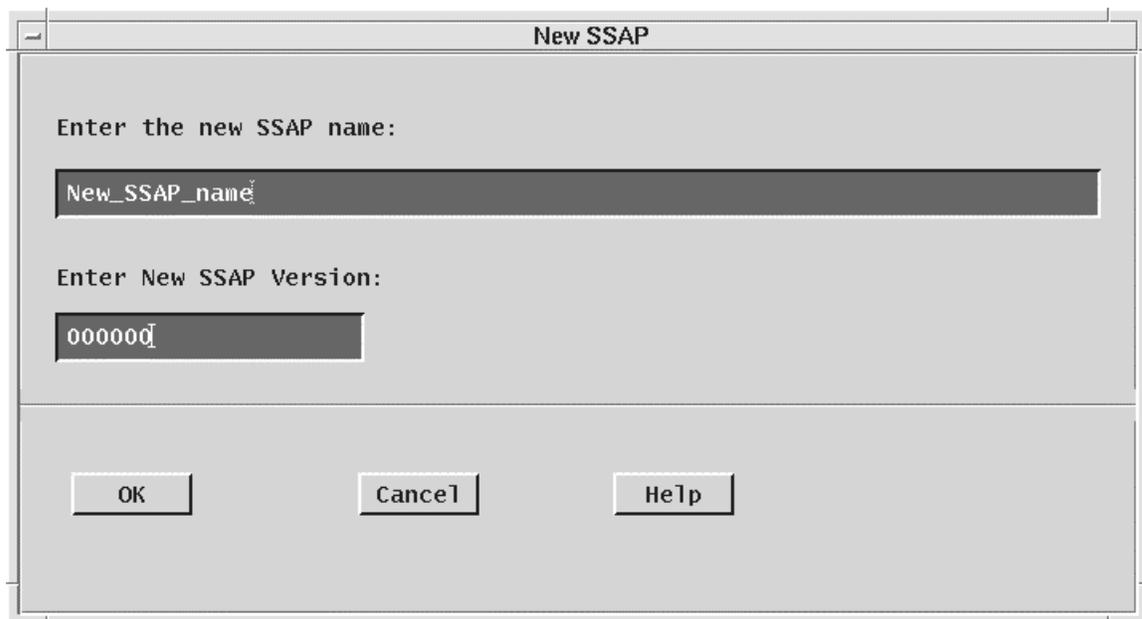


Figure 4.5.1-37. New SSAP Window

- A **Create With** button is provided as an alternative for the user to create a new SSAP starting with components of an existing SSAP file. To do this, the operator should click on an existing file, and change the parameters of the old SSAP with the name, SSAP version, and other

attributes of the new SSAP. The operator will complete the creation of the new SSAP with saving the changes to the new SSAP. Both the old and new SSAPs will retain their different identities.

- A **Submit** button allows the operator to submit the new SSAP to the Data Server for storage or update an existing SSAP.
- A **Delete** button is used by the operator for deleting a selected SSAP from the Data Server.
- A **Reset** button allows the operator to undo the changes made for the currently selected SSAP during the session.

4.5.1.2.9.6.2 File List View

Once the new SSAP has been named or an existing one has been selected, the operator can update the list of SSAP components by clicking on the **File List** tab. Selecting the **File List** tab, the operator is presented with a screen, shown in Figure 4.5.1-38.

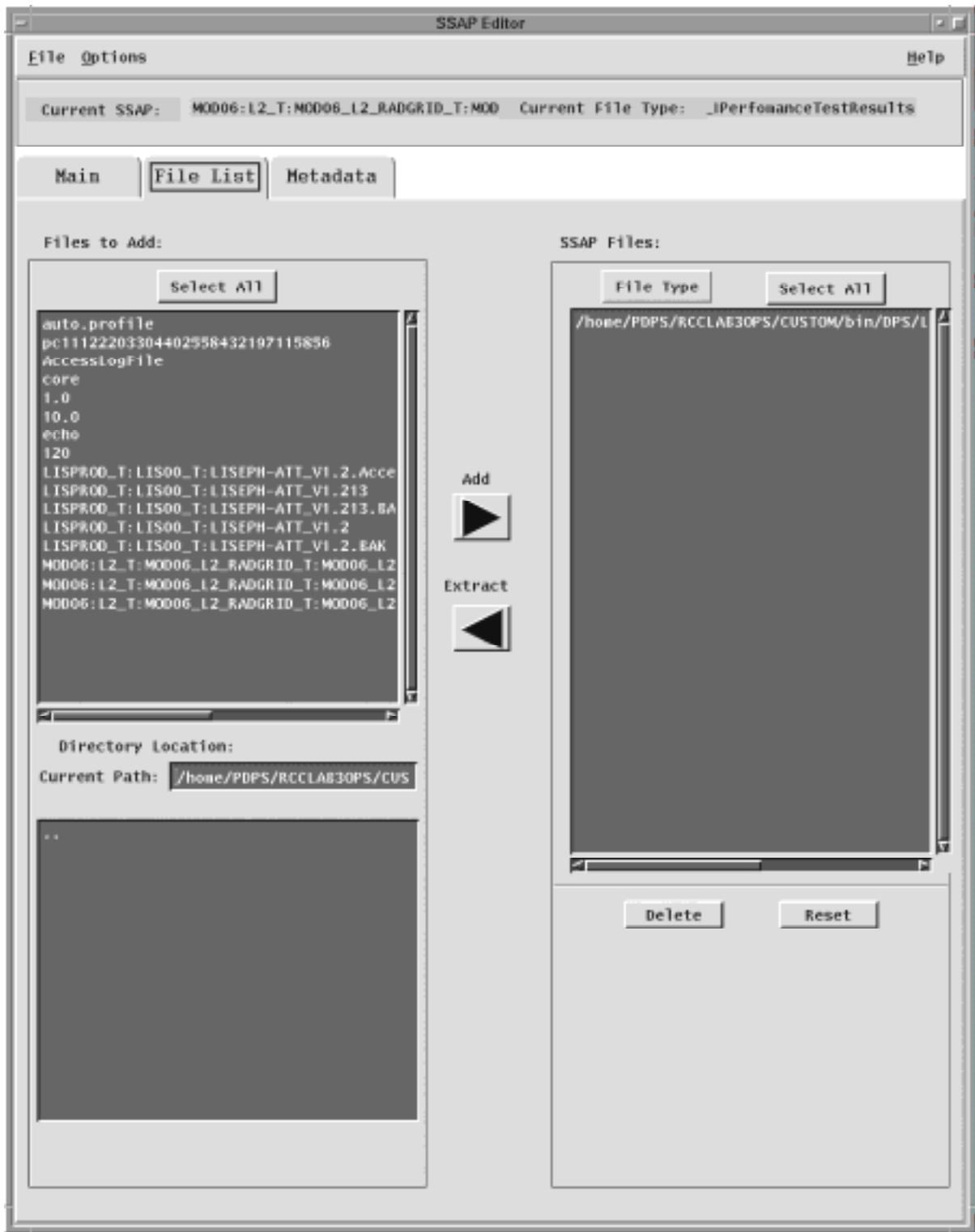


Figure 4.5.1-38. File List Pop-up

On this screen, two main windows are shown. The first window on the left side of the screen shows all the files available on the local machine and the bottom window on this side of the screen shows the directory location, showing the current path. The other window on the right side of the screen shows the files included in the SSAP. Two buttons, **Add** and **Extract**, allow for the transfer between the two aforementioned sets of files between the top two windows on the screen. Once a file is selected in the list of files of the local machine, it is listed as an SSAP file component and included in a tar file.

- By clicking on the **File Type** button, selected SSAP files can be further identified as one of the following: Algorithm Description, Change Log, Context Dialogue, Delivery List, PGE Information and Software. A **File Type** must be selected before file list changes can be made.

Other Options are offered to the operator to act on the list of SSAP files:

- The **Select All** button selects all the files in the **SSAP Files** window.
- The **Delete** button deletes the selected files; a safety window pops-up to ask the operator to confirm the delete request.
- The **Reset** button allows the operator to undo the changes just made.

4.5.1.2.9.6.3 Metadata View

For each existing SSAP, the Metadata files can be edited by the operator clicking on the **Metadata** tab. Pertinent information about the Algorithm and the PGE can be input by the operator in several text fields, shown in Figure 4.5.1-39.

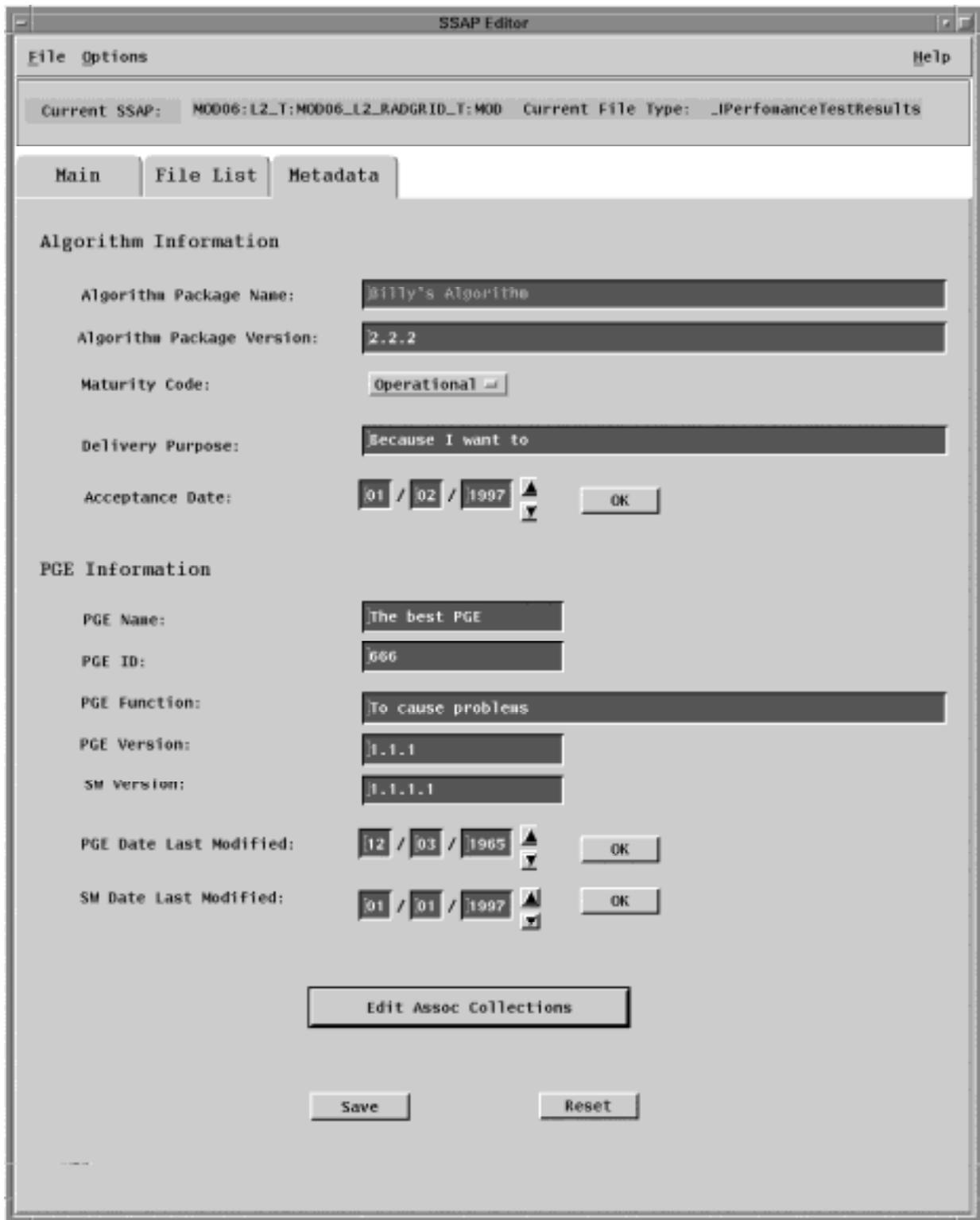


Figure 4.5.1-39. Metadata Pop-up

Some of this information is already made available by default but can be changed by the user. Descriptions of each of the fields that the operator is required to fill in are provided in the following Table 4.5.1-12

Table 4.5.1-12. SSAP Editor - Metadata Tab Field Descriptions

Field Name	Data Type	Size	Entry	Description
Algorithm Package Name	string	unlimited	user	Name of the Algorithm included in the SSAP.
Algorithm Package Version	string	unlimited	user	Version of the Algorithm included in the SSAP.
Delivery Purpose	string	unlimited	user	Reason for creating an SSAP and submitting it to the Data Server for acceptance.
Acceptance Date	date	MM:DD:YYYY	user	The date the SSAP was accepted.
PGE Name	string	unlimited	user	Name of the original PGE that was used to generate the SSAP.
PGE ID	string	unlimited	user	ID of the original PGE that was used to generate the SSAP.
PGE Function	string	unlimited	user	Metadata information on the original PGE.
PGE Version	string	unlimited	user	Version of the original PGE that was used to generate the SSAP.
SW Version	string	unlimited	user	Version of the code included in the PGE.
PGE Date Last Modified	date	MM:DD:YYYY	user	Date the PGE was last modified.
SW Date Last Modified	date	MM:DD:YYYY	user	Date the SW was last modified.

The Metadata view screen allows the operator to select the level of maturity or version of the code included in the SSAP. By clicking on the **Maturity Code** pull down menu, one of the following options can be selected: Pre-launch, Preliminary, Operational, Stable and Final.

The input of all the date fields is facilitated by the presence of ad-hoc up and down arrows that increment and decrement the values available as default or previously input by the operator.

The **Edit Assoc Collection** button allows the operator to edit the ESDT's that the original PGE is associated with. By clicking on this button, the Associated Collections Pop-up appears as shown in Figure 4.5.1-40.

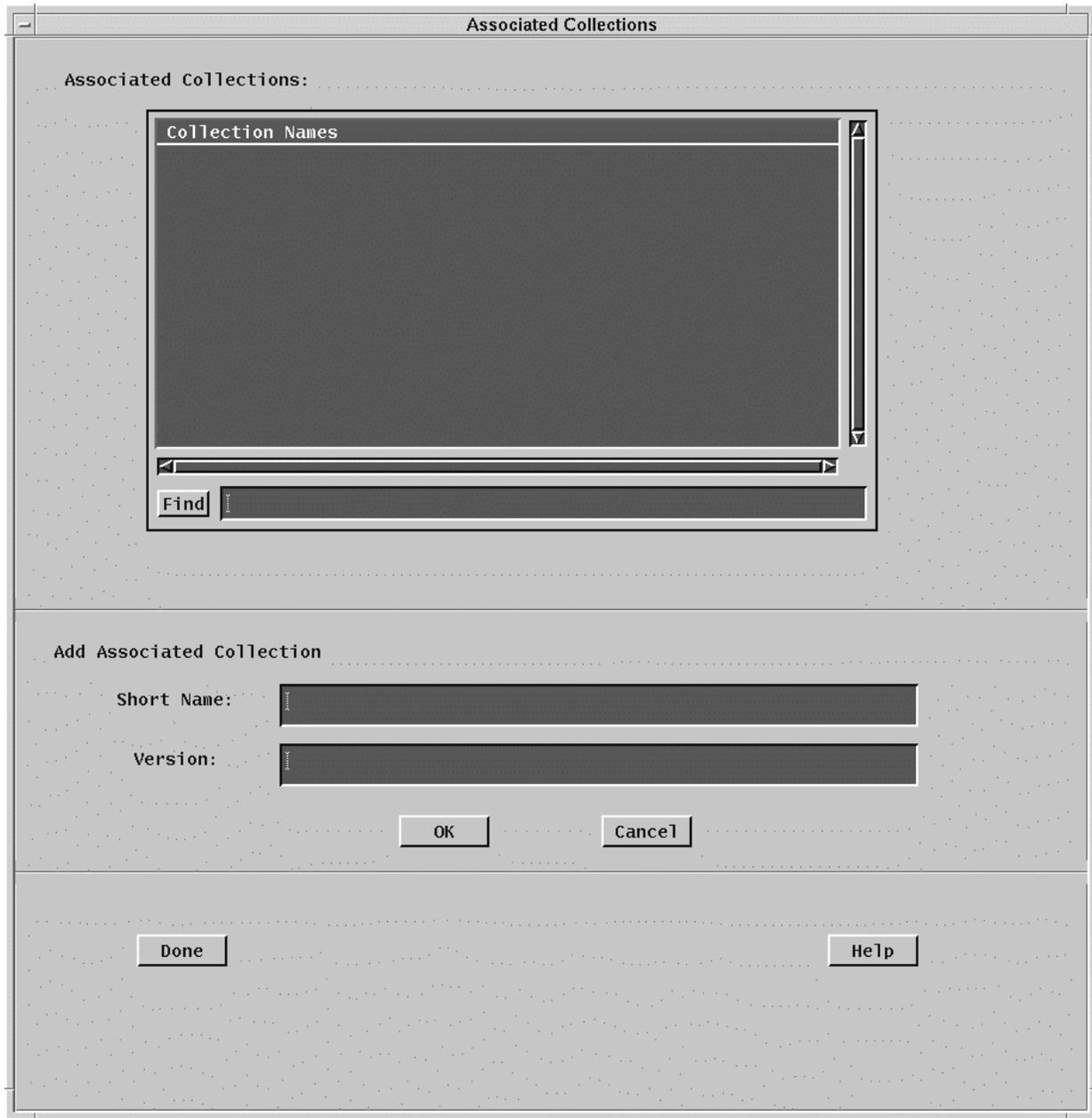


Figure 4.5.1-40. Associated Collections Pop-up

It should be noted that at least one ESDT must be associated with the original PGE in order to create a new SSAP. The Associated Collections window allows the operator to select the collection name and to input the **Short Name** and **Version** of the Associated Collections to be added. The text fields that the operator is requested to provide in this Pop-up are described in Table 4.5.1-13.

Table 4.5.1-13. SSAP Editor - Association Collection Field Description

Field Name	Data Type	Size	Entry	Description
Short Name	string	unlimited	user	Name of the Associated Collection included in the SSAP.
Version	string	unlimited	user	Version of the Associated Collection included in the SSAP.

- Once the selection and/or the addition have been completed, the operator can confirm the changes by clicking on the **Done** button. Further context sensitive help is available to the operator by clicking on the **Help** button.
- Once **Done** is clicked, the operator is returned to the Metadata screen, shown in Figure 4.5.1-39. The **File** button at the upper left hand corner of the screen is clicked, to return to the SSI&T Manager menu, and finally to quit the operation

4.5.1.2.10 Help Menu

This appears at the far right of the SSIT Manager Pop-up. Selecting the INDEX menu item brings up an index of all SSI&T Tools, as shown in Figure 4.5.1-41. Select a tool for which help is desired to bring up a help screen.

Selecting the ABOUT menu item brings up an instance of Netscape with a HTML file that provides basic information on the SSI&T Manager Tool, the Version #, etc.

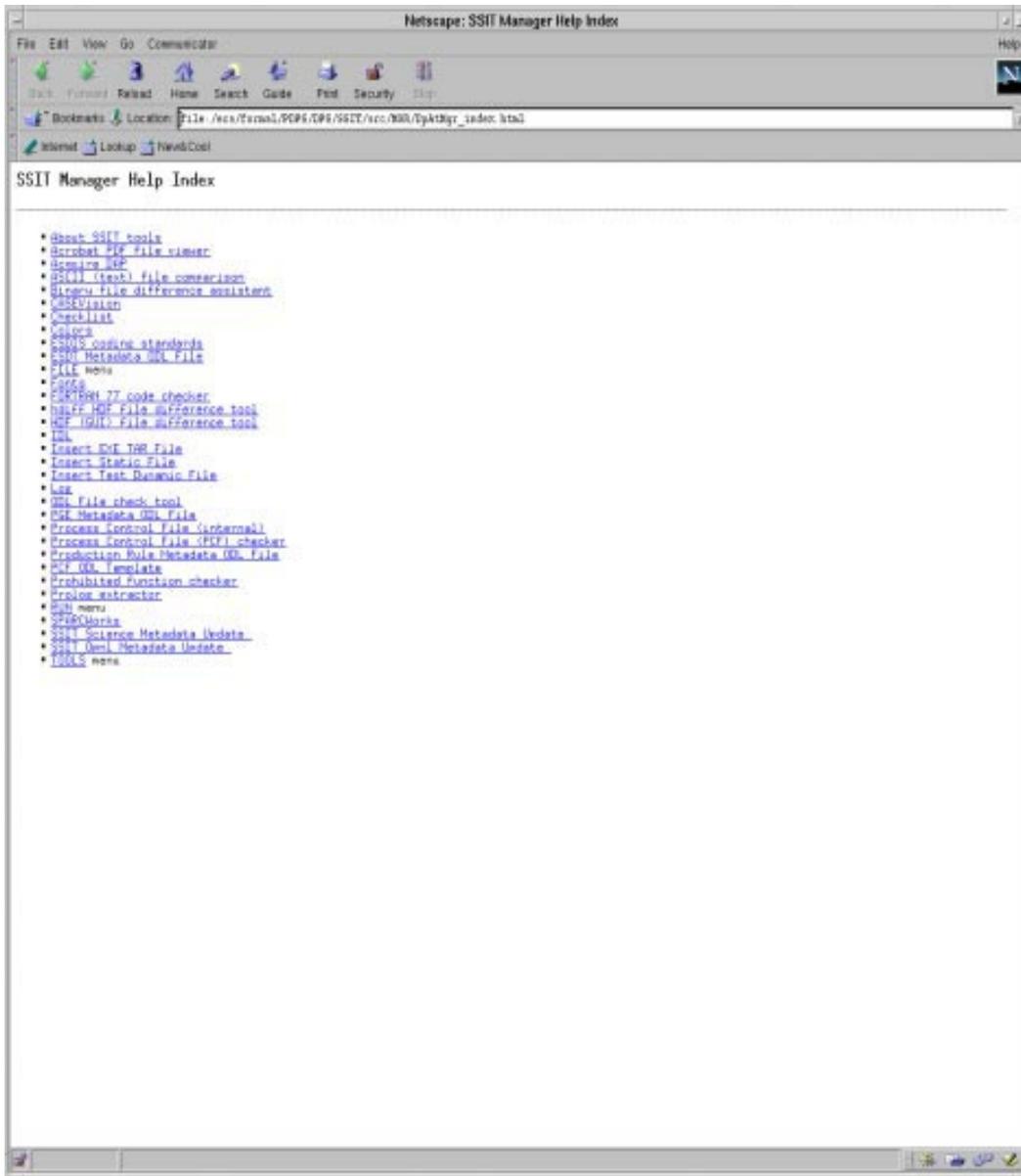


Figure 4.5.1-41. SSIT Manager Help Index

4.5.1.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled document for each product. To find the documentation for all of the COTS packages used in SSIT, refer to the ECS Baseline Information System web page, URL <http://cmdm.east.hitc.com/>.

For Ecs Custom Software tools used in SSI&T information on the operating environment, tunable parameters and environment variables refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series . The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

4.5.1.3.1 Interfaces and Data Types

The SSIT Manager exchanges data of various types through interfaces within and external to ECS. Table 4.5.1-14 lists SSIT Manager system interfaces for Version 2.0.

Table 4.5.1-14. SSIT Manager Interface Protocols

Interface (facility)	Type of Primary Interface Protocols	Type of Backup Interface Protocols	Comments
Acquire DAP	Science Data Server	none	Fixed ESDT
Insert Static	Science Data Server	none	Variable ESDT
Insert Test Dynamic	Science Data Server	none	Variable ESDT
Insert EXE TAR	Science Data Server	none	Fixed ESDT
SSAP Edit	Science Data Server	none	Fixed ESDT
Get MCF	Science Data Server	none	Fixed ESDT

4.5.1.4 Databases/ Files

The SSIT Manager uses the Unix ndbm flat-file database facility to generate the Log/Checklist database.

More importantly, the SSI&T process is closely coupled with the PDPS database. SSI&T is carried out in a different mode than production. SSI&T activities will not interfere with the production PDPS database. The PDPS database schema information is described in Section 4.7 Resource Planning of this document.

There are many files that are involved with the SSIT Manager and the associated applications. These files are summarily described in Table 4.5.1-15.

Table 4.5.1-15. SSI&T Files (1 of 2)

File Title	File Name	Purpose	Associated SSI&T Application/Tool
Installation script	\$ECS_HOME/CUSTOM/DpAtIN STALL.sh EcCoAssist	Install SSIT software	
SSI&T Internal Process Control File	\$PGS_PC_INFO_FILE (a parameter in the configuration file)	<ul style="list-style-type: none"> • SSIT Manager uses a Process Control File to determine names and locations of files • This Process Control File defines correspondence between specifications for actual files, and the logical identifiers used by the SSIT Manager to access those files • This SSI&T PCF should not be confused with the PCF used by and included with the science software 	SSIT Manager
Process Framework Configuration	\$ECS_HOME/CUSTOM/cfg/DpAt*.CFG	Enable Process Framework for PDPS database access, ECS event logging	All SSIT programs
SSIT Manager Run Menu	\$ECS_HOME/CUSTOM/data/DPS/ssit_run_menu.txt	Add items to SSIT Manager RUN menu	SSIT Manager
SSIT Manager help files	\$ECS_HOME/CUSTOM/docs/WW/DPS/DpAtMgr*.html	Online help	All tools in SSIT Manager
Sample checklist	\$ECS_HOME/CUSTOM/data/DPS/checklist.sample	SSIT Manager Checklist	SSIT Manager
ECS approved FORTRAN 77 extensions	\$FCKCNF (a parameter in the configuration file)	FORCHECK is configured to check for non-ANSI FORTRAN 77 extensions approved for use in ECS science code, according to "Data Production and SCF Standards and Guidelines".	DpAtMgrForcheck.sh
Prohibited function lists	\$ECS_HOME/CUSTOM/data/DPS/prohibitedFunctions*.txt	Suggested list of prohibited functions in science software, one for each language.	Prohibited Function Checker

Table 4.5.1-15. SSI&T Files (2 of 2)

File Title	File Name	Purpose	Associated SSI&T Application/Tool
Xresources files	In \$ECS_HOME/CUSTOM/data/DPS: <ul style="list-style-type: none"> • XDpAtMgr • Emacs • Xdiff • Xbadfunc • DpAtMgrCheckHdfFiles 	Customize appearance of Pop-ups.	Pop-ups
Binary File Difference templates	\$ECS_HOME/CUSTOM/data/DPS/DpAtMgrBinDiff*	Binary File Difference templates	DpAtMgrBinDiff
Binary File Difference example code	\$ECS_HOME/CUSTOM/data/DPS/DaacBinDiff*	Binary File Difference example code	DpAtMgrBinDiff
Sample PDPS Metadata PGE ODL file	\$ECS_HOME/CUSTOM/data/DPS/PGE_ODL.template	Examples of PDPS PGE Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata ESDT ODL file	\$ECS_HOME/CUSTOM/data/DPS/ESDT_ODL.template	Documentation of PDPS ESDT Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata TILE ODL File	\$ECS_HOME/CUSTOM/data/DPS/TILE_ODL.template	Documentation of PDPS TILE Metadata ODL Files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata PATHMAP ODL File	\$ECS_HOME/CUSTOM/data/DPS/PATHMAP_ODL.template	Documentation of PDPS PATHMAP Metadata ODL Files	DpAtPdpsDbUpdateScience (sample input)
Sample PDPS Metadata ORBIT ODL file	\$ECS_HOME/CUSTOM/data/DPS/ORBIT_ODL.template	Example of PDPS ORBIT Metadata ODL files	DpAtPdpsDbUpdateScience (sample input)

The SSIT process also uses the PDPS database. The PDPS database for Release 4 is ECS document 311-CD-106-005. The operator may have to identify individual data fields by

examination of the descriptions in the documentation. Some data may be directly accessible through the database software.

4.5.1.5 Special Constraints

None.

4.5.1.6 Outputs

Output files explicitly associated with the SSI&T custom software have been shown in the previous sections. In addition, many of the custom tools can produce output files. See the appropriate section on the custom tools for specific information.

4.5.1.7 Event and Error Messages

The SSIT Manager displays errors in a popup message box (see Appendix A for additional details). The SSIT Manager also logs error and status messages; currently these are sent to standard output. Refer to the specific vendor manuals for COTS messages.

4.5.1.8 Reports

A variety of ad-hoc and canned reports are available to the DAAC operations staff to assist in the monitoring of the activities associated with the Algorithm Integration & Test. These reports are listed in Table 4.5.1-16.

Table 4.5.1-16. Reports (1 of 2)

Report Type	Report Description	When and Why Used
Algorithm Integration & Test Management Reports	<ul style="list-style-type: none"> • provide the operations staff information on Algorithm Integration and Test application software events which have occurred • This information is available from the MSS database 	
	<ul style="list-style-type: none"> • Ad-hoc reports include: Production Requests and associated Data Processing Requests, Data Subscriptions, PGE Profiles, etc. • This information is available from the PDPS database 	These reports can be used to track modifications and provide historical information on these data objects.
PGE Profile Reports	<ul style="list-style-type: none"> • Catalogs the resource profile information associated with a PGE (e.g., generation size of PGE Output data, CPU Wall Clock Time Used, CPU actual time Used, I/O Operations). • A profile will be captured for each type of machine, i.e., Sun, SGI, etc., for which the PGE is to execute • Statistics are collected to establish standard deviations, variances, and averages of resource profile values 	These reports are used to collate information for a PGE, for a type of resource, or for a given group of PGEs used to fulfill a Production Request.

Table 4.5.1-16. Reports (2 of 2)

Report Type	Report Description	When and Why Used
I & T Activity Report	capture information about the activities which have occurred and activities which are occurring in the Algorithm Integration and Test environment	
PGE Profile Update Report	capture information to track the updates which have occurred in the PDPS Database to PGE Profiles	
PGE I&T Reports	<ul style="list-style-type: none"> • capture information on PGEs as they progress through the AI&T process. • These reports include: <ul style="list-style-type: none"> – Code Analysis Report – Standards Checker Report – File Comparison Reports – AI & T Discrepancy Reports – Inspection Reports – Integration Reports – Acceptance Reports 	These reports are to trouble shoot problems and will provide tracking and trend analysis guidance.

4.5.1.8.1 Sample Reports

These reports are described in Sections 4.7, “Resource Planning,” 4.8, “Production Planning,” and 4.9, “Production Processing” of this document.

This page intentionally left blank.

4.6 Data Ingest

ECS Data Ingest provides the software capability to acquire data by various methods and transfer the data into the ECS system. These methods include an automated transfer from prior request information, polling with or without delivery records for data which is placed at predetermined locations, and a media transfer method which includes reading tapes. The ECS Data Ingest subsystem also stores and manages request information, and provides for data preprocessing storage and insertion into the appropriate ECS storage location.

This page intentionally left blank

4.6.1 Data Ingest GUI

The Data Ingest tool allows the operator to view past ingest activities, monitor and control ingest requests, modify system and external data provider parameters, and initiate hard media ingest. Table 4.6.1-1 provides details on this tool's functions.

Table 4.6.1-1. Operator Ingest Functions

Operating Function	GUI	Description	When and Why to Use
Ingest History Log Viewing	History Log Tab	<ul style="list-style-type: none"> Displays a log that contains the results of past ECS ingest requests. Specifies search criteria (e.g., time range), the provider ID, data set name, and request status of the Ingest History Log for log display. 	As needed or for regularly scheduled Ingest performance reports or to view previously occurred Ingest activities.
Ingest Status Monitoring	Monitor/Control Tab	<ul style="list-style-type: none"> Monitors the status of all or a subset of the ingest requests that are in progress. 	As required to verify nominal system operations.
Operator Request Update	Monitor/Control Tab	<ul style="list-style-type: none"> Updates (i.e., cancel and resume) an on going ingest request on either request or granule level. 	As needed to cancel a problem request on request/granule level; or resume a suspended request on request/granule level .
Hard Media Ingest	Media Ingest Tab	<ul style="list-style-type: none"> Operator performs hard media ingest (e.g., 8mm tape) Media Ingest Session is configured on the Operator's GUI to accept the request and submit it to the ECS system 	As needed to Ingest hard media.
Modify System Parameters	Operator Tools Tab	Operator updates system parameter values including: volume threshold, request threshold, communication retry count, completed request monitor time, request monitor screen refresh time	As needed to tune the system to current or expected daily activity thresholds.
Modify External Data Provider Information	Operator Tools Tab	<ul style="list-style-type: none"> Operator updates parameter values for each external data provider Value includes volume threshold and request threshold 	As needed to tune the system to current or expected daily activity thresholds on an external data provider basis.

¹Individual DAAC policy determines the duration for which Ingest History Log information is stored and available for viewing

4.6.1.1 Quick Start Using Data Ingest

To execute the ECS Data Ingest GUI from the command line prompt, type:

>EcInGUIStart <mode>

<**mode**> is the ECS mode for the execution, e.g., OPS, TS1, etc.

4.6.1.2 ECS Data Ingest Main Screen

The Ingest Main screen, with the "Welcome to ECS Ingest GUI Interface" screen, is shown in Figure 4.6.1-1. The major functions available by clicking on tabs on this screen are:

"**Ingest Intro**" is the default tab that welcomes the user to the tool.

"**History Log**" provides operations personnel the capability to view ingest activities that are no longer active (see Section 4.6.1.2.1).

"**Monitor/Control**" provides operations personnel the capability to monitor and update ongoing ingest activities in the system (see Section 4.6.1.2.2).

"**Operator Tools**" provides operations personnel the capability to modify interactive user information by external data provider and/or modify system parameters (see Section 4.6.1.2.3).

"**Media Ingest**" provides operations personnel the capability to perform hard media ingest. The delivery record is required for media ingest (see Section 4.6.1.2.4).

There is a Menu Bar at the top of the window for getting help and activating less-frequently used secondary functions. The Menu Bar capability is available on all Ingest GUI screens. The following menus are available:

- "**File**" provides exit capability through the **exit** or **close** command and **print** and **save** capabilities, where applicable.
- "**Help**" displays general and context-sensitive help. Select "**On Help**" from the menu to get detailed help on using help. Select "**On Keys**" to get help on keyboard and mouse usage, and for general help on interacting with user interface components.

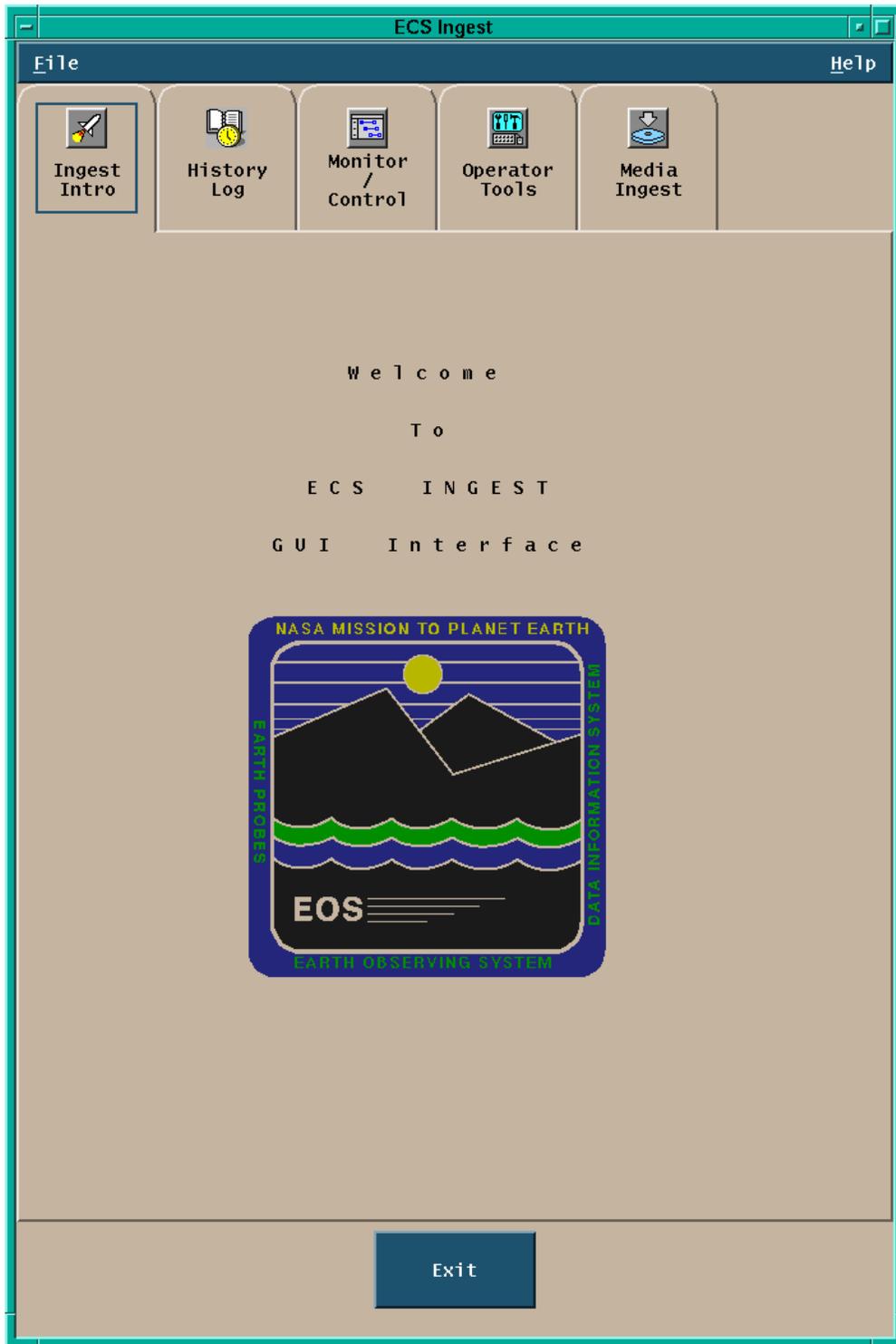


Figure 4.6.1-1. Ingest Main Screen Shown in the Ingest Intro Tab

4.6.1.2.1 History Log Tab

The History Log tab shown in Figure 4.6.1-2 provides operations personnel with the capability to view ingest requests which are no longer monitored by the Ingest system.

Operations personnel can selectively view the Ingest History Log based on the following search criteria:

- **Ingest Start and Stop Date/Time**
- **Data Provider**
- **Data Type**
- **Final Request Status**

Operations personnel can request a **Detailed Report** or just a **Summary Report** for the specified search criteria. The **Display** button populates the log window with the detailed or summary information selected. The "**History Log**" information is displayed in the history log information window.

Operations personnel can print requested History Log information by choosing the "**Print**" option from the **File** Menu.

Operations personnel can save requested History Log information by choosing the "**Save**" option from the **File** Menu. The file will be saved in the operator's default directory. This option is useful for saving Ingest History and Performance reports to a file for transmission to the SMC. History and Performance report generation is discussed in Section 4.6.1.8.

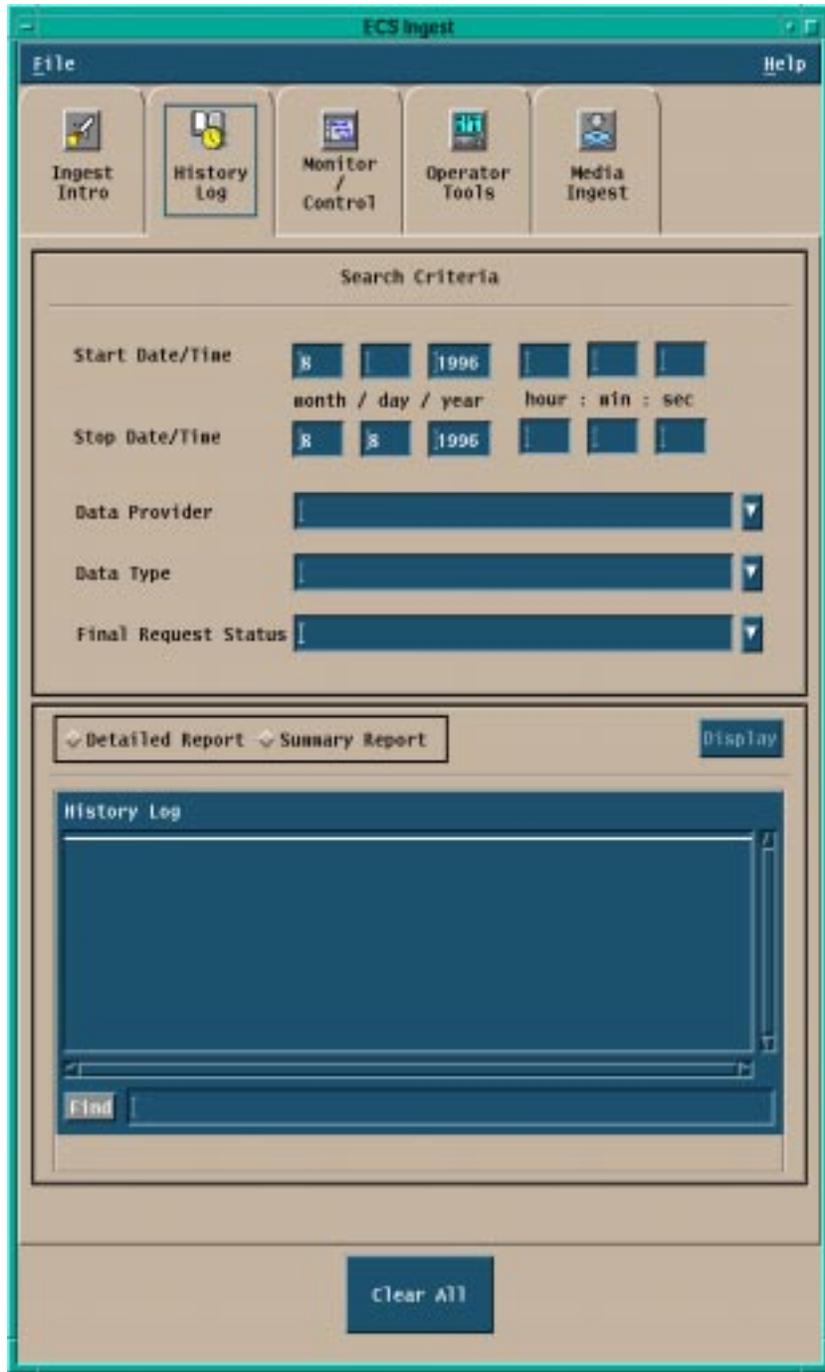


Figure 4.6.1-2. Ingest History Log Tab

Fields on the “History Log” tab are defined in Table 4.6.1-2 below.

Table 4.6.1-2. History Log Field Descriptions

Field Name	Data Type	Size	Entry	Description
Start Date/Time	Date/Time	4	Operator Selected (Default = Current Date/Time - 24 hrs.)	Entry fields for the Start Date and Time.
Stop Date/Time	Date/Time	4	Operator Selected (Default = Current Date/Time)	Entry fields for the Stop Date and Time.

Additional functionality is made available to the operator on the History Log tab through the following graphical components:

- The pull down menu “**File**”, that provides the capability to exit the GUI;
- The pull down menu “**Help**”, that displays general and context sensitive information;
- The radio buttons “**Summary Report**” and “**Detailed Report**”, that allow to toggle between the two possible optional types of report.
- The button “**Clear All**” that clears the current screen.

The selection of possible filters for the history log is automated through the availability of three lists: “**Data Provider**”, “**Data Type**”, and “**Final Request Status**” from which the operator can chose a number of possible options.

4.6.1.2.2 Monitor/Control Tab

Clicking on the Monitor/Control tab brings up the **Ingest Status Monitoring/Request Control** tab as shown in Figure 4.6.1-3. This tab provides operations personnel with the capability to monitor ongoing ingest activities in the system and can alert the operator if a request has passed the requested expiration date. The Control feature allows the operator to cancel a request on either request or granule level. It also allows a suspended request to be resumed at either request or granule level.

The Ingest Monitor/Control tab is divided into four regions:

- Search By allows operations personnel to view all or selected ingest requests in the system.
- View Selection and Action Type button. The tab defaults to show the graphical view for the request information. If the text view button is selected, the text view region is shown as indicated in Figure 4.6.1-4.
- Information display -- allows the operator to display information in two ways based upon the selection of **Graphical View** or **Text View**
- Control Types – designed to provide operations personnel with the capability to update ongoing ingest activities in the system. Update services include **resume** and **cancel**.

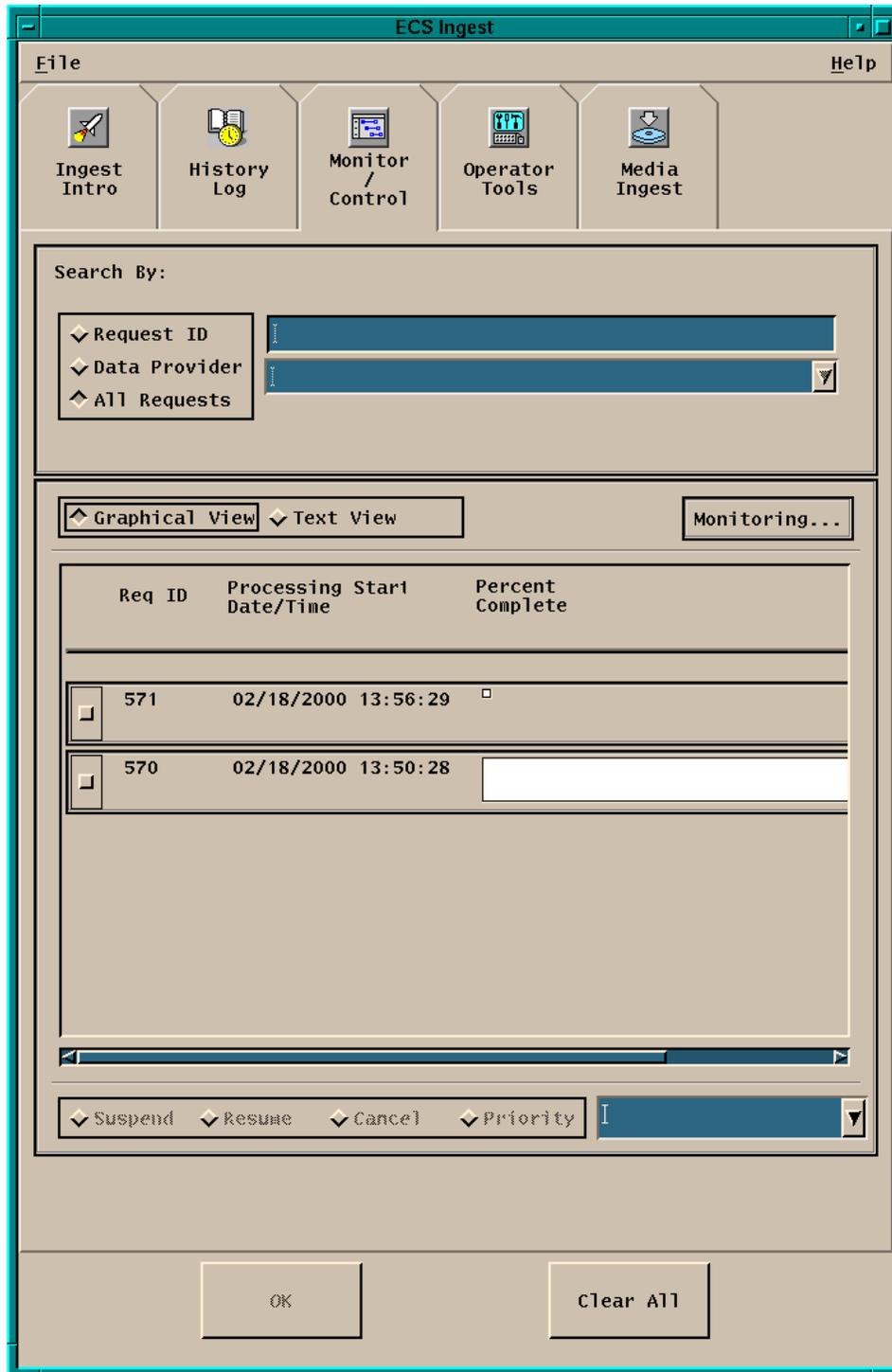


Figure 4.6.1-3. Ingest Monitor/Control Tab (Graphical View)



Figure 4.6.1-4. Ingest Monitor/Control Tab (Text View)

Table 4.6.1-3 describes the fields available on the Monitor/Control tab.

Table 4.6.1-3. Ingest Monitor/Control Tab Field Descriptions

Field Name	Data Type	Size	Entry	Description
Request ID	Integer	N/A	Operator Selected	Searches for a specific request currently in the ingest system.

Additional functionality is offered to the operator through the following graphical components:

- The “**File**” pull down menu that provides **exit** capability;
- The “**Help**” pull down menu that displays general and context-sensitive help;
- The “**Monitoring.../Control...**” event label is used to display the current state of the underlining process. “**Monitoring...**” lets the operator view the list of the current requests, with information including the **Request ID**, the **Processing Start Date/Time**, and the **Percent Complete** in the Graphical view and the **Request Id**, the **Status**, the **Data Provider**, the **Ingest Type**, the **Priority**, and the **Start Date** in the Text View. “**Control...**” allows the operator to cancel or resume a request or a granule from a list;
- The pull down list “**Data Provider**”, which allows the operator to select from a list of valid external data providers;
- The “**All Requests**” toggle button that displays all requests currently in the system;
- The “**Suspend**” radio button that suspends the processing of an ongoing ingest request (inactive);
- The “**Resume**” radio button that resumes the processing of a previously suspended ingest request on either request or granule level;
- The “**Cancel**” radio button that aborts the processing of an ongoing ingest request on either request or granule level;
- The “**Priority**” radio button that is used in conjunction with a pulldown list of allowed priority levels allows the operator to assign a priority order to an ingest request (inactive).

4.6.1.2.3 Operator Tools Tab

Clicking on the Operator Tools tab brings up the Operator Tools window shown in Figure 4.6.1-5. This tab provides the means to set system-level and data provider-level threshold and setup information for simple throttling of the ingest process. In particular, operations staff may control both the number of requests to be processed concurrently and the amount of data to be ingested

concurrently (for the entire site and for each data provider individually). Note: All system-level and data provider-level thresholds and setup information will have initial values at the time of delivery.

The **Operator Tools** tab is divided into three subtabs:

- **Modify External Data Provider / User Information subtab** allows setup of thresholds and user information for external data providers, including authorized science users who interactively request data ingest (see Figure 4.6.1-5)
- **Modify System Parameters subtab** allows setup of thresholds and system information on a system-wide basis as shown in Figure 4.6.1-7.
- **File Transfer subtab** allows setup of external node information as shown in Figure 4.6.1-8.

See Tables 4.6.1-4, 4.6.1-6, and 4.6.1-7 below for a description of fields that appear, respectively, in the **External Data/User Information**, **Modify System Parameters**, and **File Transfer** subtabs.

In addition to the above input fields the three Operator Tools tabs include other graphical elements that provide additional functionality to the operator.

“**File**” and “**Help**” pulldown menus allow, respectively, for exiting capability and displaying of context sensitive help information.

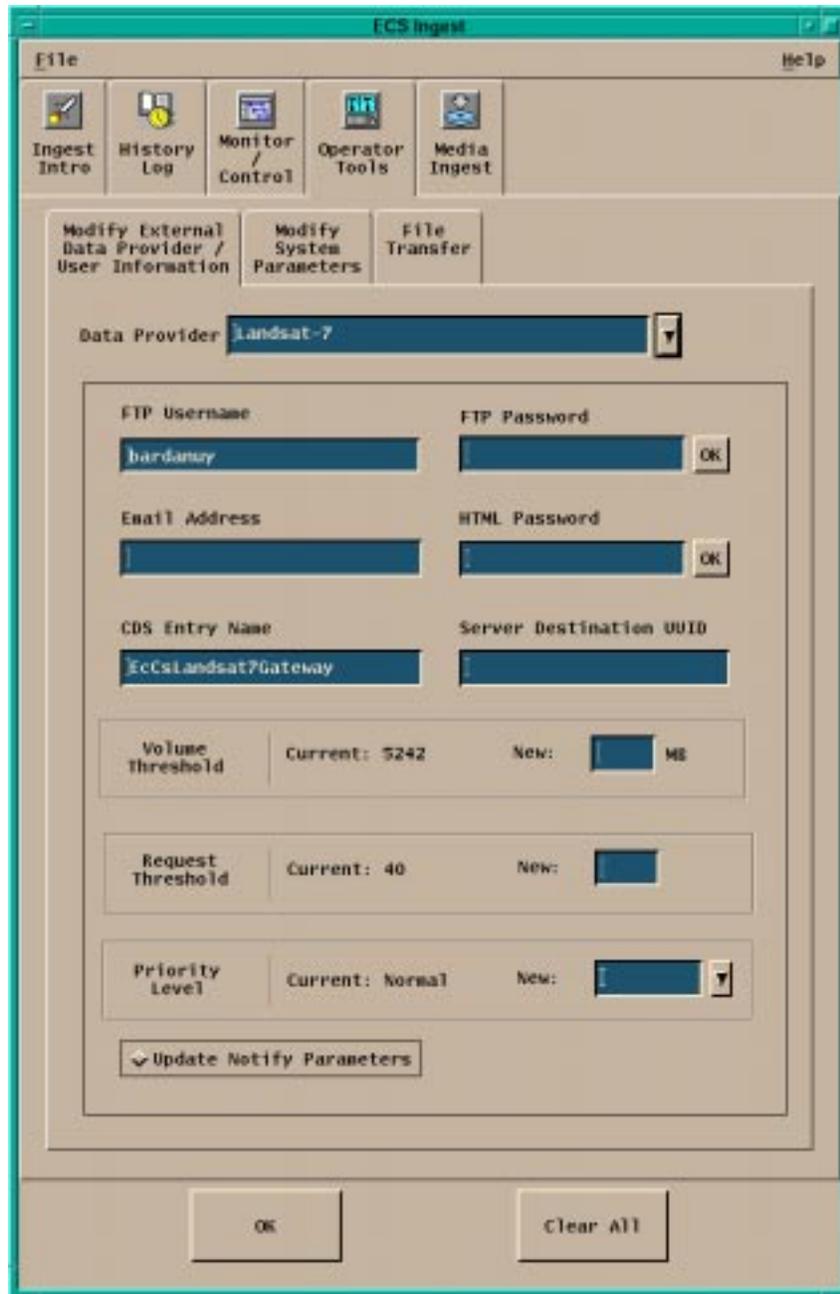


Figure 4.6.1-5. Operator Tools - External Data Provider/User Information Subtab

The Modify External Data Provider/User Information subtab includes a pull down list that allows for the selection of valid External **Data Provider** IDs. The operator clicks on **OK** to implement the changes or **Clear All** to delete all the modifications operated so far.

**Table 4.6.1-4. Operator Tools - External Data/ User information Tab
Field Descriptions**

Field Name	Data Type	Size	Entry	Description
FTP Username	Text Field	10	Operator Selected (Default = current value or blank, if no current value exists)	Username of the external data provider at a remote node holding data.
FTP Password	Text Field	30	Operator Selected (Default = blank)	<ul style="list-style-type: none"> • Provides an input entry field for FTP Password • The FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)
Email Address	Text Field	255	Operator Selected (Default = blank)	Email address of external data provider.
HTML Password	Text Field	30	Operator Selected (Default = blank,)	<ul style="list-style-type: none"> • HTML password for authorized science users • The HTML password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)
CDS Entry Name	Text Field	255	Operator Selected (Default = current value or blank, if no current value exists)	Provides the CDS entry for the given data provider.
Server Destination UUID	Text Field	36	Operator Selected (Default = current value or blank, if no current value exists)	Provides the UUID for the given data provider.
Volume Threshold	Integer	n/a	Operator Selected	Provides the current ingest volume threshold for the given data provider and an input area to set a new value.
Request Threshold	Integer	N/A	Operator Selected	Provides the current ingest request threshold for the given data provider and an input area to set a new value.

The **Priority Level** of the Data Provider can be changed using a list of valid priority levels (default is VHIGH).

If the toggle button **Update Notify Parameters** is selected in the **External Data Provider/User Information** subtab the **Notify Parameter** pop-up appears as shown in

Figure 4.6.1-6. Table 4.6.1-5 describes the fields contained in this pop-up. With the **OK** and **Cancel** buttons on the Notify Parameter, the operator can accept or delete the changes made.

The image shows a dialog box titled "Notify Parameters". It has a light blue header bar with the title. The main area is white and contains several input fields. The first field is "Notify Type" with the text "Buffer" entered. Below it are two columns of fields: "Notify FTP Node" and "Notify FTP Directory" in the first row, and "Notify FTP Username" and "Notify FTP Password" in the second row. Each of these four fields is empty. At the bottom left of the dialog is an "OK" button, and at the bottom right is a "Cancel" button. There is also a small "OK" button located at the end of the "Notify FTP Password" input field.

Figure 4.6.1-6. Update Notify Parameters Pop-up

Table 4.6.1-5 describes the fields on the Update Notify Parameters Pop-up.

Table 4.6.1-5. Update Notify Parameters Field Descriptions

Field Name	Data Type	Size	Entry	Description
Notify Type	Text Field	10	Operator Selected (Default = current value or blank, if no current value exists)	Provides Notify Type for the given data provider.
Notify FTP Node	Text Field	255	Operator Selected (Default = blank)	Provides Notify FTP node for the given data provider.
Notify FTP Directory	Text Field	255	Operator Selected (Default = blank)	Provides Notify FTP directory for the given data provider.
Notify FTP Username	Text Field	10	Operator Selected (Default = blank)	Provides Notify FTP username for the given data provider.
Notify FTP Password	Text Field	30	Operator Selected (Default = blank)	<ul style="list-style-type: none"> • Provides an input entry field for Notify FTP Password • The Notify FTP Password is NOT reflected to the screen when the operator inputs information (and is encrypted prior to storage)

Figure 4.6.1-5 is the Modify System Parameters subtab.

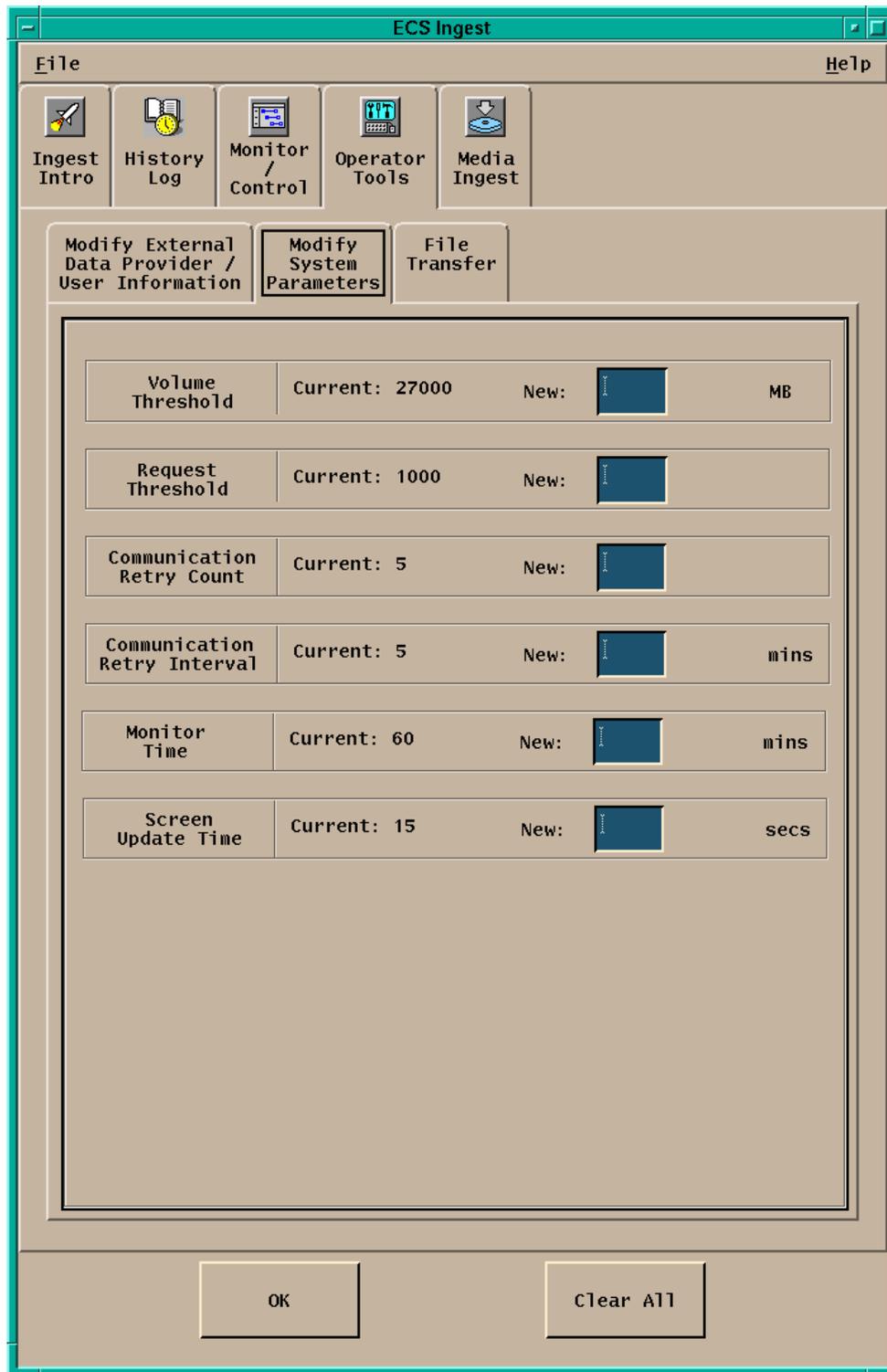


Figure 4.6.1-7. Operator Tools - Modify System Parameters Subtab

Table 4.6.1-6 describes the fields on the Modify System Parameters subtab.

**Table 4.6.1-6. Operator Tools Modify System Parameters
Field Descriptions**

Field Name	Data Type	Size	Entry	Description
Volume Threshold	Integer	n/a	Operator Selected (Default = current value)	Provides the current system volume threshold and an input area to set a new value.
Request Threshold	Integer	N/A	Operator Selected (Default = current value)	Provides the current system setting for ingest maximum requests and an input area to set a new value.
Communication Retry Count	Integer	N/A	Operator Selected (Default = current value)	Provides the current Communication Retry Count and an input area to set a new Communication Retry Count.
Communication Retry Interval	Integer	N/A	Operator Selected (Default = current value)	Provides the current system setting for communication retry interval in minutes and an input area to set a new value.
Monitor Time	Integer	N/A	Operator Selected (Default = current value)	<ul style="list-style-type: none"> • Provides the current amount of time (in minutes) that a completed request is retained on the monitoring display prior to removal. • Provides an input area to set a new monitor time
Screen Update Time	Integer	N/A	Operator Selected (Default = current value)	<ul style="list-style-type: none"> • Provides the current system setting for screen update interval in seconds and an input area to set a new value

The **OK** and **Clear All** buttons will allow operators to accept or delete changes to the System Parameters.

Figure 4.6.1-8 is the File Transfer subtab.

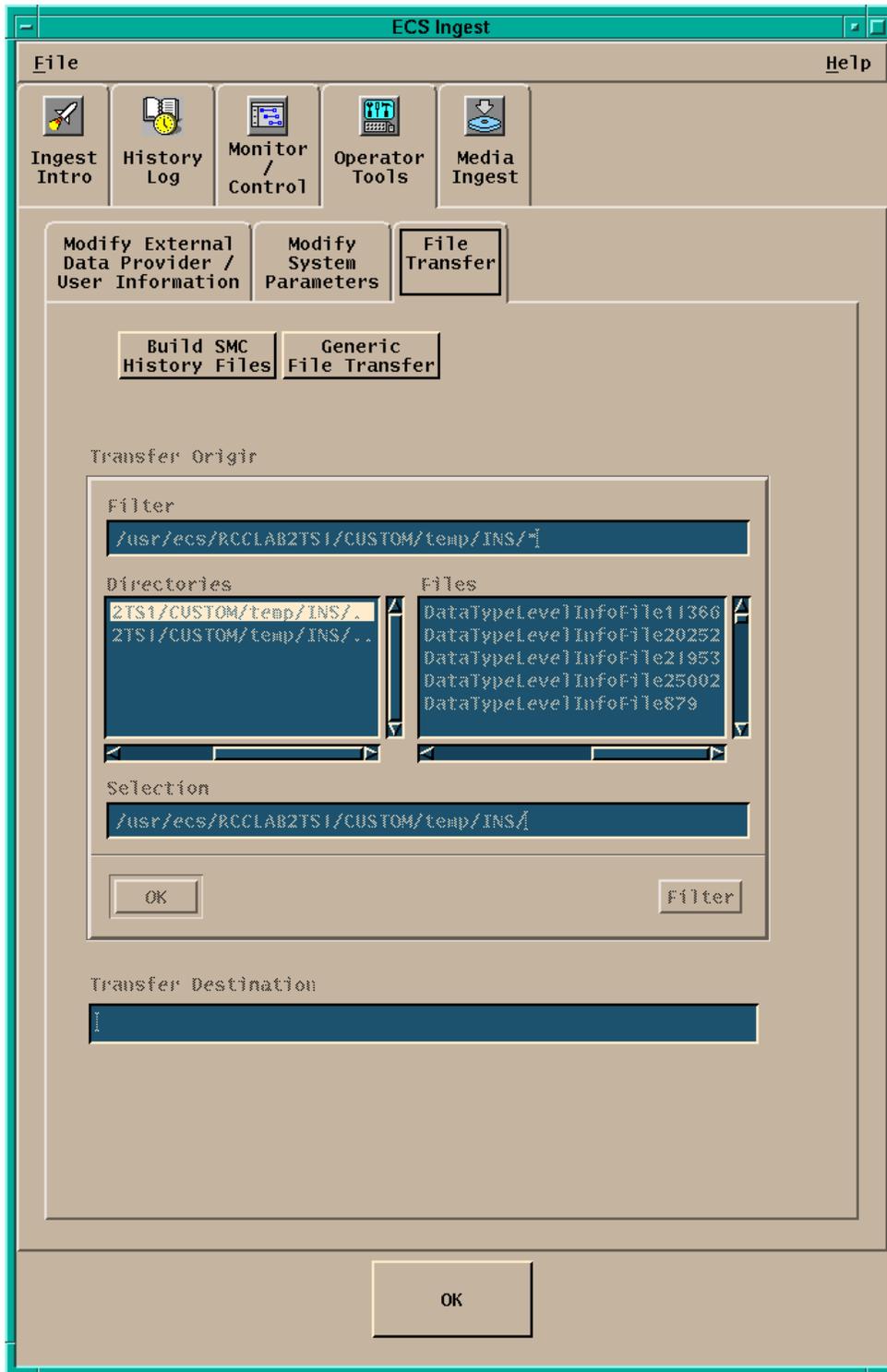


Figure 4.6.1-8. Operator Tools - File Transfer Subtab

Table 4.6.1-7 describes the fields on the File Transfer subtab.

Table 4.6.1-7. Operator Tools - File Transfer Field Descriptions

Field Name	Data Type	Size	Entry	Description
Filter	Text Field	unlimited	Operator Selected	Provides a default directory for file selection
Selection	Text Field	unlimited	Operator Selected	Provides the name of file for transfer
Transfer Destination	Text Field	unlimited	Operator Selected	Provides the file transfer destination
Directories	character	unlimited	System generated	list of directory available for filtering
Files	character	unlimited	System Generated	list of files available for filtering

The buttons at the top of the window allow to specify if the operation objective is to perform a **Generic File Transfer** or to **Build SMC History Files**. With the **OK** button the operator initiate the desired file transfer.

4.6.1.2.4 Media Ingest Tab

Clicking on the Media Ingest tab brings up the **Ingest Media** window as shown in Figure 4.6.1-9. This tab provides DAAC operations personnel with the capability to perform media ingest (e.g., specifying media for ingest, such as 8mm tape).

The Ingest Media Ingest tab is divided into three regions:

- Media Type – allows the operator to select three types of media (8mm tape, 4mm tape, and D3 tape)
- Data Provider and Media Volume ID – allows the operator to specify media stacker information (if a stacker is available for the media type), the media provider ID, and media ID.
- Data Delivery Record File Location– allows the operator to specify the location of the media data delivery record file, which is required for media ingest. The file could either be embedded in the hard media or be made available electronically. If not embedded on the hard media, the Delivery Record file must be in an ECS disk location. The external data provider must connect with that location via ftp prior to delivering the hard media to ECS.

A different GUI is presented if the media type chosen requires a stacker. See Figure 4.6.1-9 for an example of a GUI without a Stacker. See Figure 4.6.1-10 for an example of a GUI with a Stacker ID. Table 4.6.1-8 describes the fields of this tab.

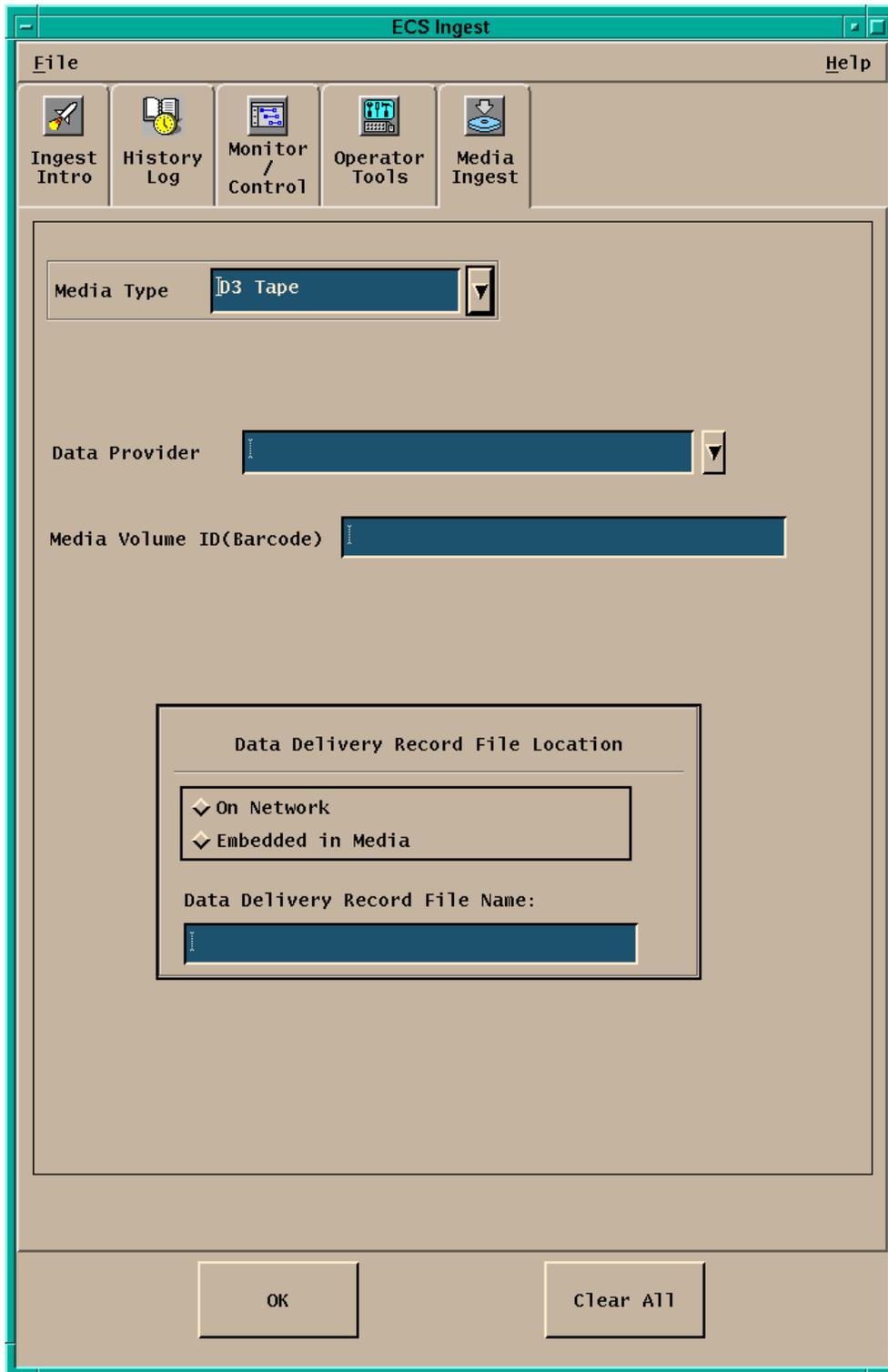


Figure 4.6.1-9. Ingest Media Tab (Without Stacker ID)

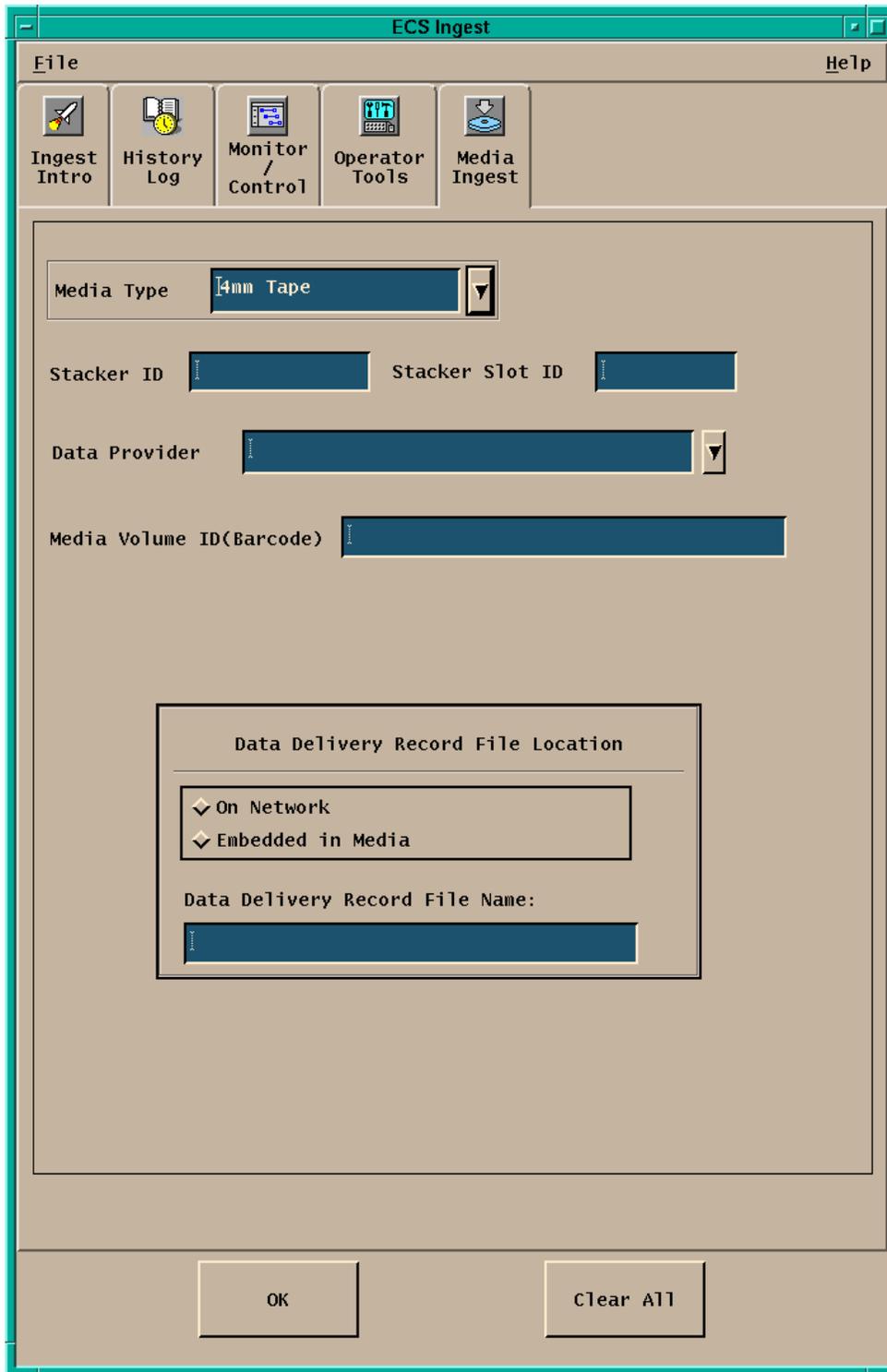


Figure 4.6.1-10. Ingest Media Tab Requiring Stackers

Table 4.6.1-8 describes the fields on the Ingest Media tab.

Table 4.6.1-8. Ingest Media Field Descriptions

Field Name	Data Type	Size	Entry	Description
Media Volume ID	integer	N/A	Operator Selected	Media Volume ID input
DDR File Name	character	unlimited	Operator Selected	Name of the Data Delivery Record file
Stacker Slot ID	integer	N/A	Operator Selected	Provides the location (slot ID) of the hard media in the stacker (Presented if Media Type chosen requires a stacker)
Stacker ID	integer	N/A	Operator Selected	Identifies the media stacker in which the media is to be inserted(Presented if Media Type chosen requires a stacker)

Media Type and **Data Provider** can be selected from lists of valid names. The location of the Data Delivery Record file can be selected through by toggling the **On Network** and **Embedded in Media** buttons.

With the **OK** and **Clear All** buttons the Operator can accept start the Ingest session or delete the entries input in any of the fields.

4.6.1.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC. These documents can be found in the Technical Documents of the ECS BIS.

The following table 4.6.1-9 identifies the supporting products the Data Ingest depend upon in order to function properly in the ECS environment:

Table 4.6.1-9. ECS Data Ingest Product Dependency

Product Dependence	Protocol Used	Comments
INS GUI	X-11	via client libraries
INS GUI and ReqMgr	DCE	via RPC call

4.6.1.4 Databases

The ECS Data Ingest Tool interface with the Ingest Database. For details about the design and schema of the Ingest Database refer to 311-CD-103-005, *Ingest Database Design and Schema Specifications*.

4.6.1.5 Special Constraints

None.

4.6.1.6 Outputs

Operations staff have access to three output logs, described in Table 4.6.1-10.

Table 4.6.1-10. Outputs

Output	Description and Format
Ingest history log	The Ingest history log contains summary information about ingest request status. Stored in Sybase.
MSS event log	The MSS event log contains critical events of interest to DAAC operations staff. Stored in MSS.
Ingest event log	The Ingest event log contains critical and detailed events of interest to DAAC operations staff. Stored in a flat file.

4.6.1.7 Event and Error Messages

The Ingest interface issues both status and error messages to the event log file. Error messages fall under the following categories: common messages, history log messages, ingest media ingest messages, ingest request controller messages, and ingest threshold controller messages. Both event and error messages are listed in Appendix A.

4.6.1.8 Ingest Reports

In addition to ad hoc ongoing request status displays discussed above, the Ingest subsystem provides the standard reports described in Table 4.6.1-11.

Table 4.6.1-11. Standard Ingest Production Reports

Report Type	Report Description	When and Why Used
Ingest Request History Report	<p>The report supplies operations staff with a view of ingest request completion performance. It provides a detailed log of the ingest requests in the reporting period (including requester, data source, data type, the times of various ingest events such as request receipt and completion, data volume, etc.).</p> <p>The report also provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories.</p>	<p>The report may be generated for specified time periods and executed on a regular basis (based on site policy).</p>
Ingest Request Performance Report	<p>The report supplies operations staff with a view of ingest performance. It provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories.</p>	<p>The report may be generated for specified time periods and executed on a regular basis (based on site policy).</p>
Ingest Granule Performance Report	<p>The report supplies operations staff with a view of ingest granule-level performance. It provides summary statistics for the reporting period, such as completed vs. unsuccessful requests, backlog development, average ingest volumes and processing times broken down by various categories..</p>	<p>The report may be generated for specified time periods and executed on a regular basis (based on site policy).</p>

4.6.1.8.1 Sample History Reports

An example Ingest Request History Report is shown in Figure 4.6.1-11.

Req ID	Data Provider	Compl . Status	Ingest Type	Start Time	End Time	Total Granules	Success Granules	Data Volume	File Count	Transfer Time	Pre - Proc Time	Archive Time	Priority	Restart Flag
00100	TSDIS	OK	PD	01-05-97 12:31:06	01-05-97 13:51:16	1	1	112.5	5	53.6	3.2	23.4	----	---
.....	-----	-----

Figure 4.6.1-11. Sample Ingest Request History Report

An example Ingest Request Performance Report is shown in Figure 4.6.1-12.

Data Provider	Total Requests	Total Request Errors	Granules		Files		Size		Transfer Time		Pre - Proc Time		Archive Time	
			Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
SDPF	112	2	24	112	62	336	106.5	114.5	51	105	7	23	12	212
TSDIS												

Figure 4.6.1-12. Sample Ingest Request Performance Report

An example Ingest Granule Performance Report is shown in Figure 4.6.1-13.

Data Provider	Data Type	Total Granules	Total Granules with Errors	Files Avg Max		Granule Vol. Avg Max		Transfer Time Avg Max		Pre - Proc Time Avg Max		Archive Time Avg Max	
SDPF	CER00	112	2	62	336	106.5	114.5	51	105	7	23	12	212
SDPF

Figure 4.6.1-13. Sample Ingest Granule Performance Report

This page intentionally left blank.

4.7 Resource Planning

This section describes the Resource Planning tools used by DAAC operators. These tools are accessible through Resource Planning GUIs.

This page intentionally left blank.

4.7.1 Resource Planning

Resource planning is accomplished by operations through a Resource Editor tool that is used to define the resources that will be allocated to run production data and a Resource Scheduler GUI that is used to create resource reservations. The Resource Scheduler can also display resource reservations on a time line.

The Resource Editor GUI can define hardware resources such as disks and host computers ('real computers') as well as virtual resources such as 'virtual computers' which are the collections of CPUs and disks. The list of resources can be created from scratch or they can be initially obtained from a resource configuration in MSS.

A resource plan is defined by a set of approved resource reservations. The Resource Scheduler GUI is used by the resource planner to create, validate and approve reservations until a conflict-free plan is achieved. At this point the reservations will be "committed" and the reservations become ground events. The Ground Event activities include maintenance, test, and training. .

Table 4.7.1-1 provides a high level summary of the activities of the Resource Planning GUIs.

Table 4.7.1-1. Common ECS Operator Functions Performed with Resource Planning GUIs (1 of 2)

Operating Function	GUIs Involved	Description	When and Why to Use
Ingest /create and modify resource configuration.	Resource Editor, Virtual Computer Details, Disk Details, Hardware Details, String Details, Real Computer Details, Autosys Details	Obtains from the MSS CM system a list of system resources which is used for resource planning and to edit/add to these resource items.	<ul style="list-style-type: none"> • To define system resources • To add and modify resources
Enter a resource reservation request.	Resource Scheduler, Reservation Edit/Definition, Resource Selection, Intervals Selection	The resource planner enters a request for a resource reservation.	To define the resource reservation for testing or maintenance
Review resource reservation requests in the system	Resource Scheduler, Reservation Edit/Definition, Resource Selection, Intervals Selection	The resource planner reviews the resource reservations that have been made.	To support resource allocation process, to aid in preparing a resource reservation request, or to inspect the use of system resources

Table 4.7.1-1. Common ECS Operator Functions Performed with Resource Planning GUIs (2 of 2)

Operating Function	GUIs Involved	Description	When and Why to Use
Commit approved resources.	Resource Scheduler	The resource planner allocates a resource to a particular activity.	<ul style="list-style-type: none"> To register the reservation as the ground event
Review Timeline.	Resource Reservation Planning Master Timeline	A timeline oriented view of resource allocation for all configured system resources.	To aid in planning resource use within the system.

4.7.1.1 Quick Start Using Resource Planning

There are two main applications associated with Resource Planning:

- **Resource Scheduler GUI** is used to enter a new resource reservation, to modify an existing resource reservation request, to approve a resource reservation, to commit approved resource reservations, to view a timeline format display of all approved resource reservations, and to delete an existing reservation.
- **Resource Editor GUI** is used to add or delete resources or modify the characteristics of resources.

To invoke the Resource Editor GUI from the command line, enter:

EcPIRReStart <mode> <MSGSVR_ID>

<mode> is the ECS mode for the execution.

The <MSGSVR_ID> is the TBS Server Identifier.

To execute the Resource Planning Scheduler from the command line, type:

EcPIRpSiStart <mode> <MSGSRV_ID>

<mode> is the ECS mode for the execution.

The <MSGSVR_ID> is the TBS Server Identifier.

4.7.1.2 Resource Scheduler Main Screen

The Resource Scheduler GUI, depicted in Figure 4.7.1-1, appears when the Resource Scheduler Icon is clicked. This GUI presents a list of the resource reservations entered into the system, with one line of information per resource reservation. The list can be filtered by activity type by clicking on the button displayed next to “Activity Type” at the top left of the screen. The selection button lists each of the activity types defined to resource planning, e.g., ‘Testing’. Selecting an activity from this list limits the display to resource reservations with that type of activity specified.

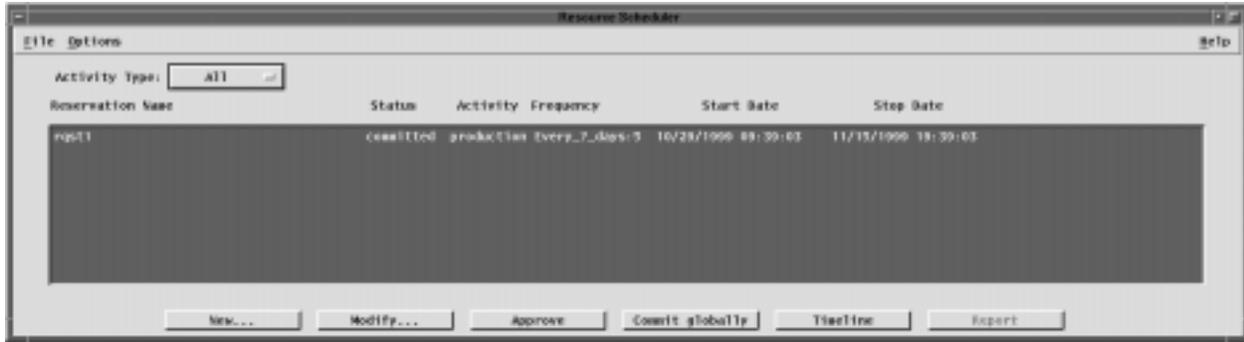


Figure 4.7.1-1. Resource Scheduler GUI

There are several buttons at the bottom of the GUI which affect the status of the reservations. The functions associated to these buttons are also accessible from the menu bar. These buttons and their actions are:

- **New ...** - Used to enter a new resource reservation request. This brings up the Resource Reservation Request Edit/Definition GUI described in the section 4.7.1.1.3.
- **Modify ...** - Used to edit or review the details of an existing resource reservation request. First select a resource reservation entry from the list in the Resource Planning GUI, then click on 'Modify...' This brings up the Resource Reservation Request Edit/Definition GUI for the selected resource reservation
- **Approve** - Used to indicate that all reviews associated with the resource planning process have taken place and there are no objections to the resource usage as described by the request. Clicking on this button checks if there is any conflict between this resource reservation and other reservations. Error messages are displayed if conflicts are detected
- **Commit** - Commits all approved resource reservations. At this point, the information related to the activity and its reserved resources is accessible by the production planning software.
- **Timeline** - Displays a timeline oriented view of the resource plan in the Resource Reservation Plan Timeline GUI. This GUI is discussed further in Section 4.7.1.1.4.
- In addition, on the menu bar, the pulldown menus provide the following capabilities.
- 'File' Pulldown:
 - **New** - Same as the 'New' button on the GUI - used to enter a new resource reservation request.
 - **Open** - Same as the 'Modify...' button on the GUI - used to update information for an existing resource reservation request.
 - **Delete** - deletes a resource reservation request.

- **Exit** - exits the application.
- 'Options' Pulldown:
- **Timeline** - Same as the 'Timeline' button on the GUI - used to display a timeline oriented view of the resource plan.

4.7.1.2.1 Resource Reservation Request Edit/Definition GUI

From the Resource Planning GUI, a user enters a new resource reservation, or reviews an existing resource reservation. Either selection displays the Resource Reservation Request Edit/Definition GUI shown in Figure 4.7.1-2.

Resource Reservation Request Edit/Definition - Modify

Request Name:

Edited Date:

Originator:

Sponsor:

Activity: Priority:

Description:

Start Day as "MM/DD/YYYY" Start Time as "HH:MM:SS"

Stop Day as "MM/DD/YYYY" Stop Time as "HH:MM:SS"

Frequency:

Rejected
 Validated
 Status:

Comments:

Figure 4.7.1-2. Resource Reservation Request Edit/Definition GUI

Table 4.7.1-2 describes the fields of the Resource Reservation Request Edit/Definition GUI.

**Table 4.7.1-2. Resource Reservation Request Edit/Definition GUI
Field Description**

Field Name	Data Type	Size (# of characters)	Entry	Description
Request Name	ASCII characters	<= 40	user input, required	A name for the resource request
Edited Date	Date	40	system generated	Date of resource request entry
Originator	ASCII characters	<= 30	system generated	Userid of the user entering the resource request.
Sponsor	ASCII characters	<= 30	user input	Name of the individual designated to review and validate the Resource Request for completeness, etc.
Priority	Integer	3	user provided	The priority for the activity
Description	ASCII characters	30	User provided, optional	User description of the activity for which the resource is required
Start Day	Date	10	User provided, required	The start date of the resource request.
Start Time	Time	8	User provided, required	The start time of the resource request.
Stop Day	Date	10	User provided, required	If a the reservation is to be repeated at some frequency (see below) the stop date specifies the end date for the repeated request.
Stop Time	Time	8	User provided, required	The stop time of the resource request.
Status	ASCII characters	<= 20	system generated	Displays the status of the request, e.g., 'approved', 'validated', etc.
Comment	ASCII characters	unlimited	User provided, optional	User comment on the resource reservation request.

Selecting the '**Resources ...**' button will display the Resources Selection GUI, shown in Figure 4.7.1-3, that displays two lists, one containing all of the available resources and the other listing the resources that the operator selects for incorporation into the resource reservation. The user selects the desired resource by means of the arrow buttons. The name of the resource request is displayed, once the reservation is saved.

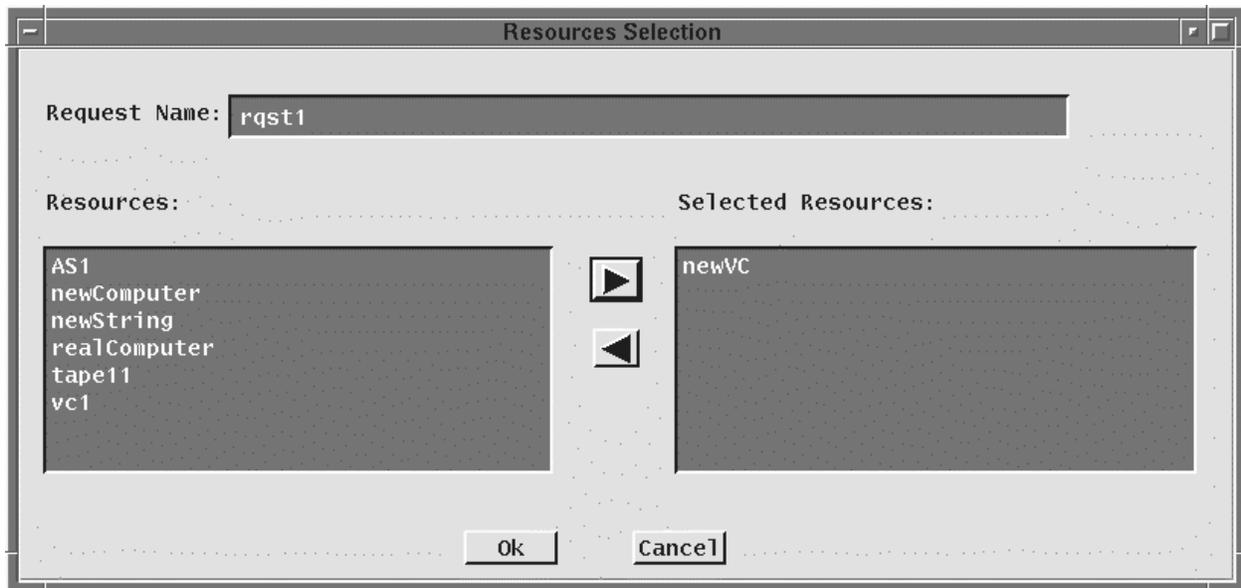


Figure 4.7.1-3. Resources Selection GUI

In the time period specification region of the Resource Reservation Request Edit/Definition GUI, the user can specify the frequency of occurrence of a repeating resource request. Several options for expressing the frequency are available in the ‘**Frequency**’ selection list box combined with a text field that provides a qualifier. The available frequency options are listed in Table 4.7.1-3 below.

Table 4.7.1-3. Frequency Qualifiers for Resource Reservation Request Edit/Definition GUI

Frequency Specifier	Text Qualifier	Result
Once	--	The default. Resource reservation covering the period from the start time and stop time for the start date specified.
Monthly	--	Resource reservation for every month on the start day of the month, repeated until the end date as specified.
Daily	--	Resource reservation for every day, between the start date and end date, for the start time and end time specified.
Every M-W-F	--	Resource reservation for every Monday, Wednesday, and Friday, between the start date and end date, for the start time and end time specified.
Every M-through-F	--	Resource reservation for every Monday through Friday, between the start date and end date, for the start time and end time specified.
Every T,Th	--	Resource reservation for every Tuesday & Thursday, between the start date and end date, for the start time and end time specified.
Weekends	--	Resource reservation for every Saturday & Sunday, between the start date and end date, for the start time and end time specified.
'Every_?_days'	n-days	Resource reservation for every n-days, between the start date and end date, for the start time and end time specified.

The **'Intervals ...'** button provides the mechanism to tailor a 'Frequency' based request by overriding selected intervals. If the user selects the 'Intervals ...' button, the Intervals Selection GUI shown in Figure 4.7.1-4 is displayed, providing two lists. The window on the left displays a list of dates selected through the 'Frequency' button. The window on the right shows dates that the operator selects for exclusion from the initial interval. Items are moved from one list to the other by selecting the item and using the arrow keys. The name of the resource request is also displayed.

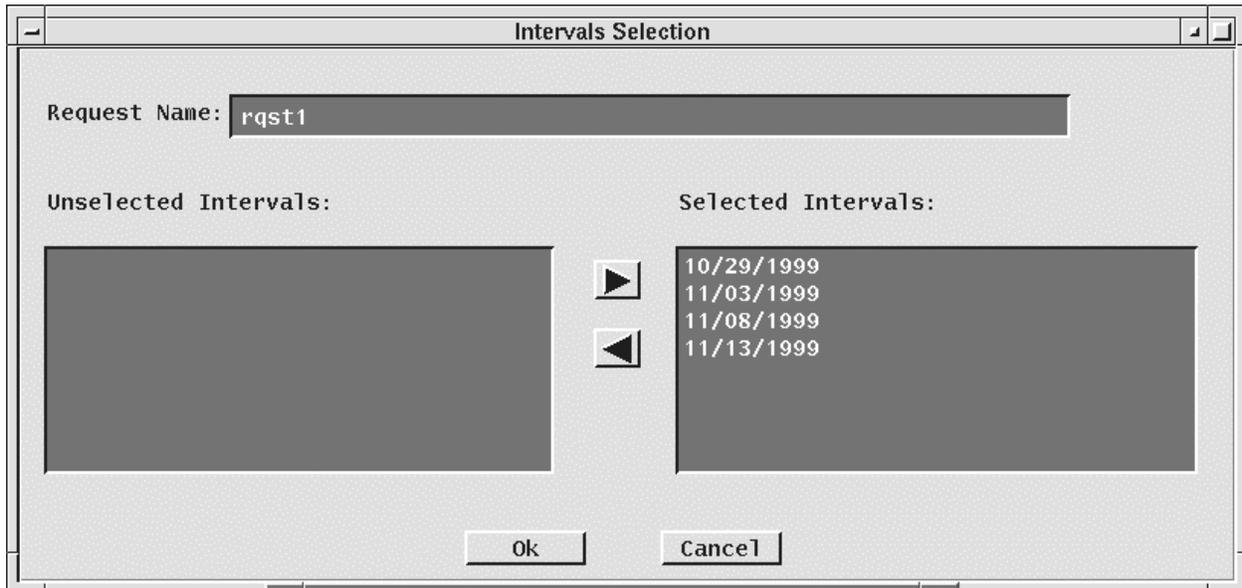


Figure 4.7.1-4. Intervals Selection GUI

4.7.1.2.2 Resource Reservation Planning Master Timeline GUI

From the Resource Planning GUI, the user can select the **'TimeLine'** button to view a timeline format display of all approved resource reservations, similar to the example displayed in Figure 4.7.1-5.

The Resource Reservation Planning Master Timeline GUI represents a set of resources, arranged along the left side of the GUI, and some period of time as indicated across the top edge of the GUI. The use of a resource over a period of time is represented by one or more 'resource reservation' bars across the GUI. A bar represents a time period during which a resource reservation has been approved for the resource. Each bar has the name of the resource reservation. At those times where a reservation has not been placed against a resource, that resource is allocated to a default activity (which can vary by resource). For example, science processing computers are used for science processing unless a reservation has been placed against them. Scroll bars allow scrolling up and down through the full list of resources and left and right in time. A select list of time span viewing options (e.g., 24-hours, 48-hours) at the lower left of the screen is available for selecting the time span of interest.

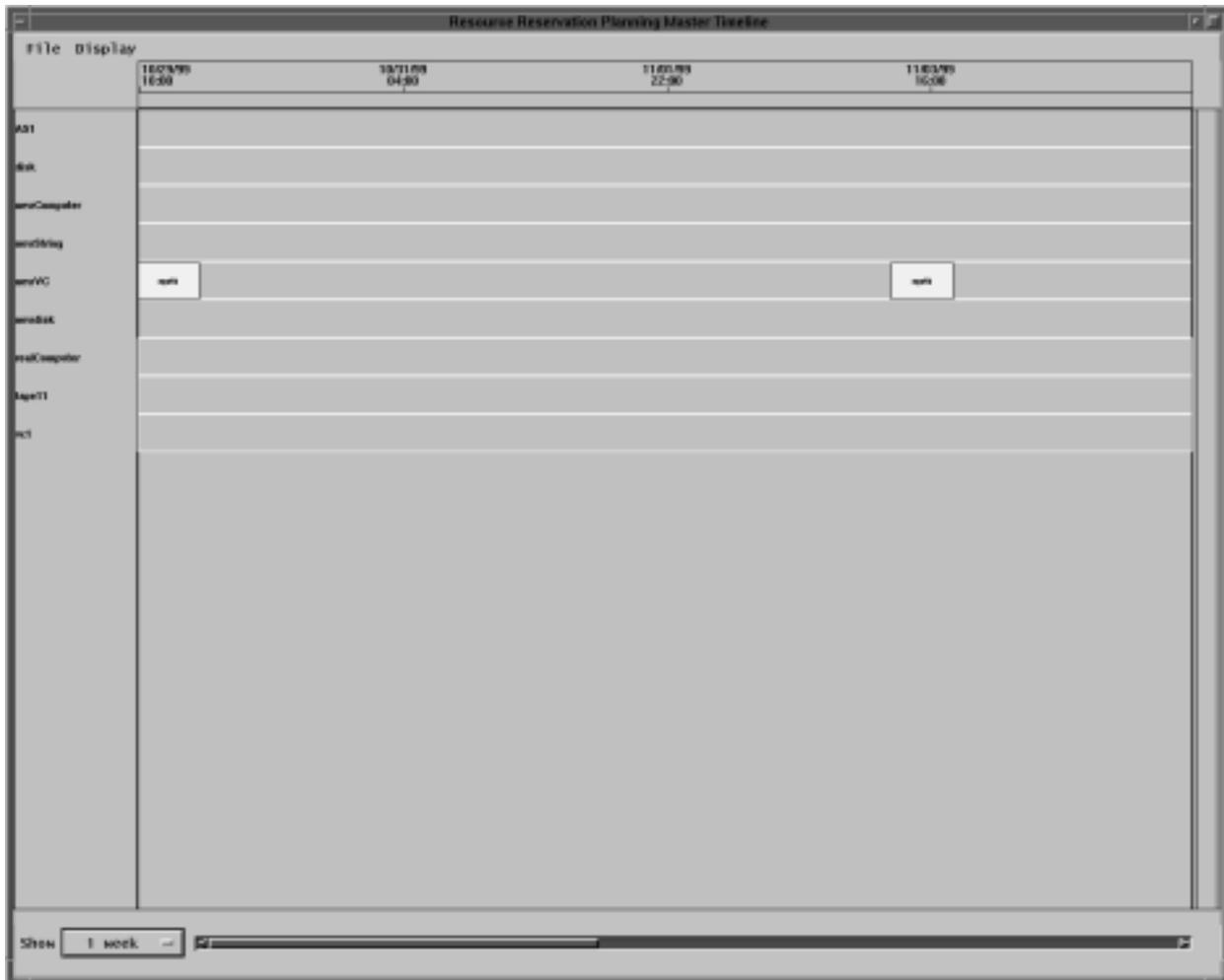


Figure 4.7.1-5. Resource Reservation Planning Master Timeline GUI

The capabilities to adjust some of the timeline display features (e.g., colors) are available through the pulldown menus located at the top of the GUI. These capabilities include:

- **'File'** Pulldown:
 - **Quit** - Exits this application and returns to the Resource Planning GUI
- **'Display'** Pulldown:
 - **Change Color** - Changes the color setup of the display.
 - **Change Resource** - Selects or filters the resources displayed in the timeline.
 - **Change Timescale** - Changes the plan window start and stop time

4.7.1.2.2.1 Report Generator GUI

Note, this functionality will be made available in future releases.

Figure 4.7.1-6. Report Generator GUI (Future Release)

4.7.1.3 Resource Editor GUI

The resource list may be created from scratch or be initialized from the MSS-managed configuration list, through a 'fetch baseline' operation. Resources may be added to or deleted from the resource list without affecting the MSS configuration list. This is useful, for example, for identifying resources that will be available in the future for the purposes of planning. The Resource Editor GUI, shown in Figure 4.7.1-8, is the tool used for adding resources or modifying the characteristics of existing resources. On start-up, the Resource Editor GUI displays a list of the resources known to the resource planning tool. To add a new resource the operator first selects a resource type using the **Resource Type** selection button. Then the operator clicks on the '**New...**' button. The characteristics of an existing resource can be modified clicking on the resource in the resource list and clicking on '**Modify...**'. A resource can be removed by selecting it and clicking on '**Delete**'.

The retrieval of configuration information is a two step process. First a Tivoli job is run (see Section 4.2.3) that generates a file of configuration information used by Resource Planning. This job is run by pressing the '**Fetch Baseline**' button. When the Tivoli job has completed, press the '**Load Baseline**' button to extract the needed information from the Tivoli generated file and load it into the PDPS database.

The GUIs for the definition of specific resource are discussed in the following subsections.

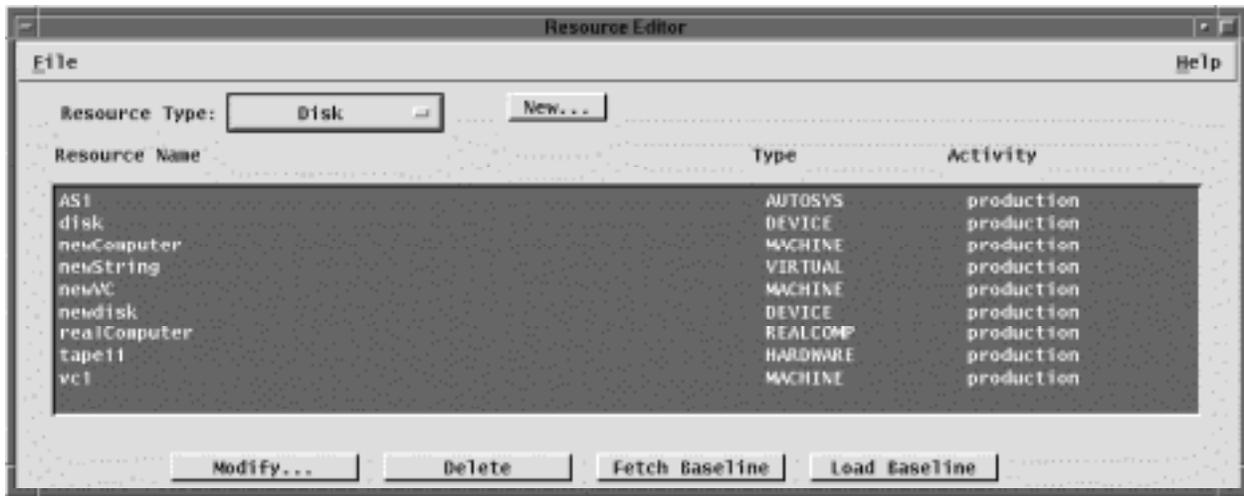


Figure 4.7.1-8. Resource Editor GUI

In addition, the 'File' pulldown located on the menu bar provides an 'Exit' option to leave the application.

4.7.1.3.1 Hardware Details GUI

The Hardware Details GUI, shown in Figure 4.7.1-9, is used to define the characteristics of generic hardware resources at the DAAC, such as communications equipment.

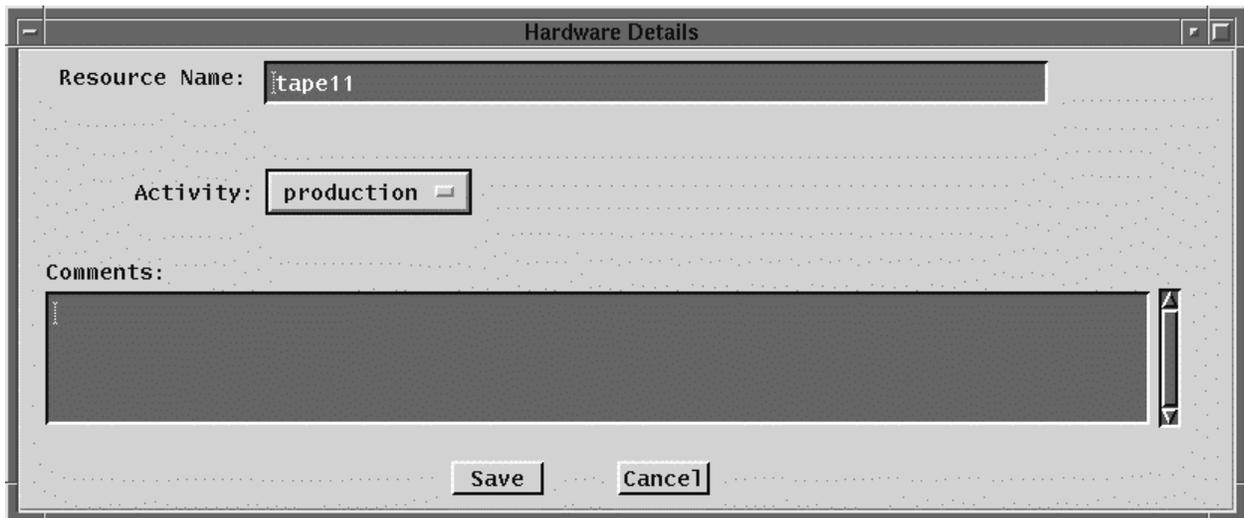


Figure 4.7.1-9. Hardware Details GUI

The characteristics of the other fields are described in Table 4.7.1-4.

Table 4.7.1-4. Hardware Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Comments	ASCII characters	unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.2 Disk Details GUI

The Disk Details GUI, shown in Figure 4.7.1-10, is used to define the characteristics of disk partition type resources.

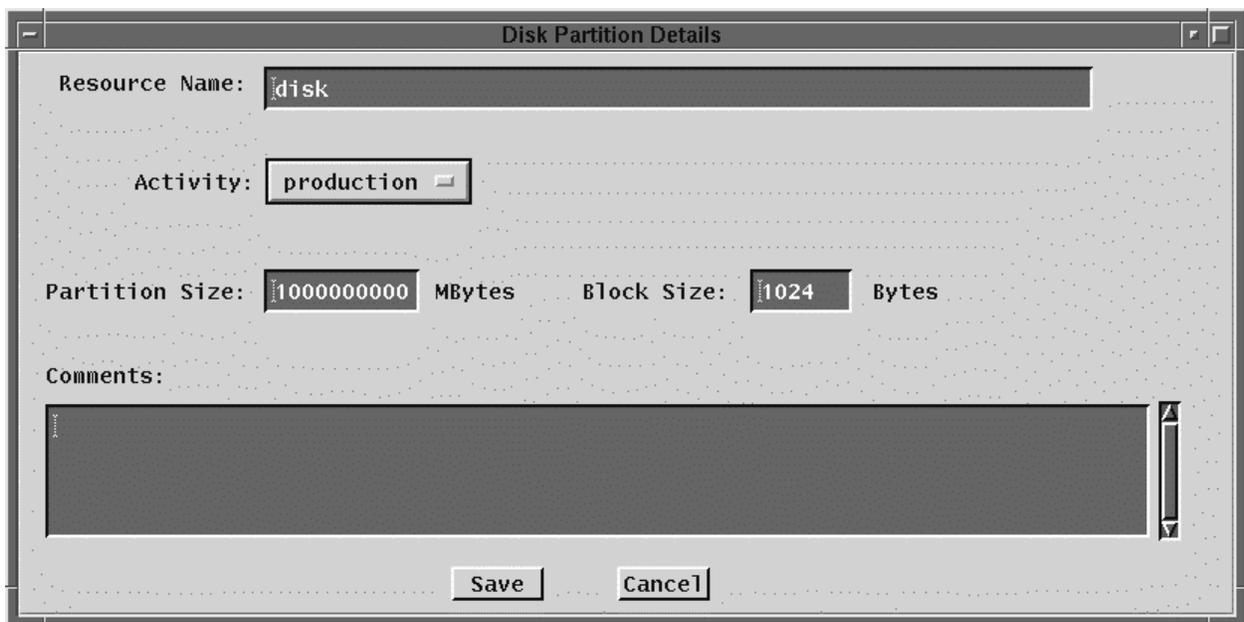


Figure 4.7.1-10. Disk Details GUIs

The characteristics of the other fields are described in Table 4.7.1-5.

Table 4.7.1-5. Disk Resource Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Partition Size	float	<= 10	User provided, required.	The size of the disk partition, in bytes.
Block Size	Integer	<= 6	User provided, required.	Block size used for the disk in bytes.
Comments	ASCII characters	unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.3 Virtual Computer Details GUI

The Virtual Computer Details GUI shown in Figure 4.7.1-11 is used to define the characteristics of virtual computer resources .

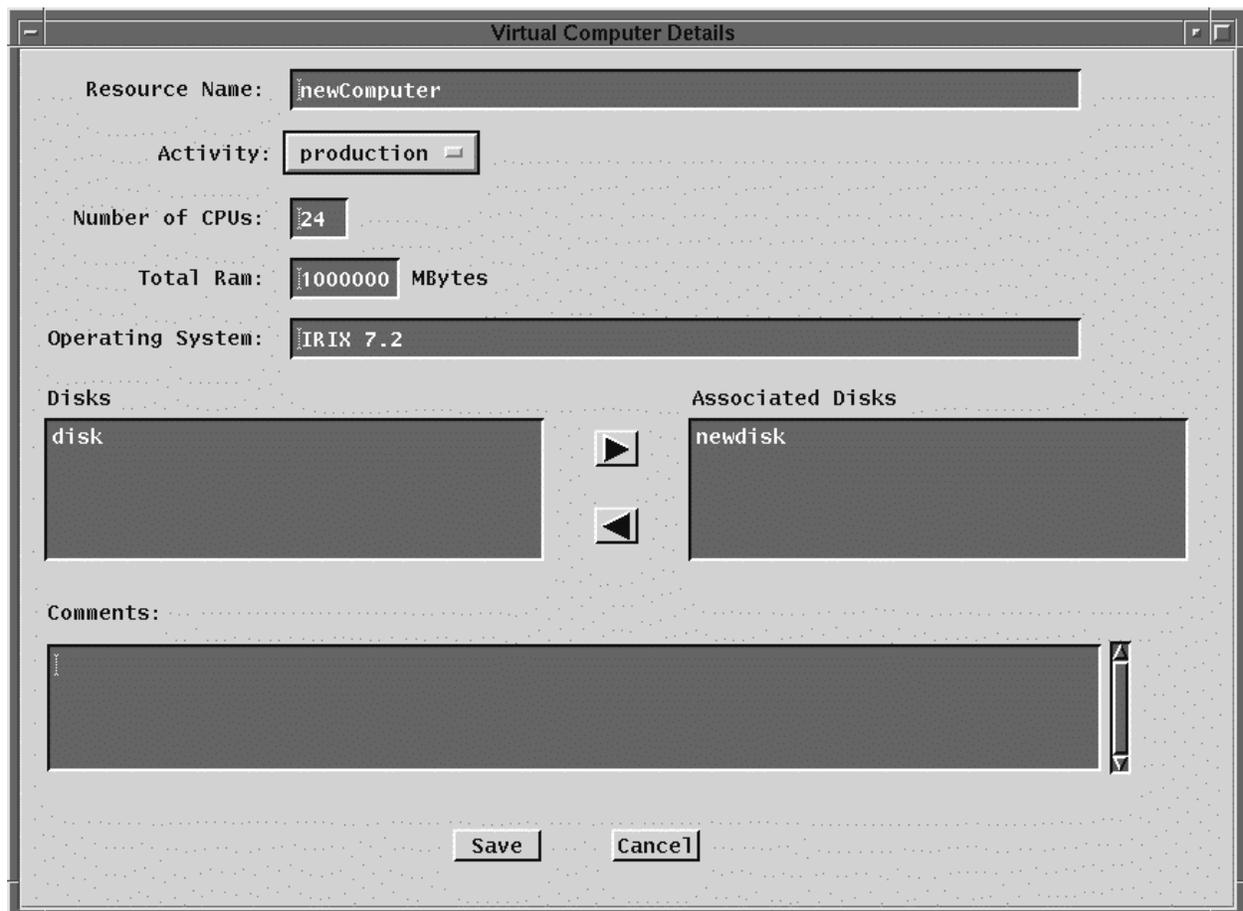


Figure 4.7.1-11. Virtual Computer Details GUIs

The characteristics of the other fields are described in Table 4.7.1-6. .

Table 4.7.1-6. Virtual Computer Details GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required	A user defined name for the computer.
Number of CPUs	Integer	<= 3	User provided, required.	The number of CPUs within the computer.
Total RAM	Integer	<= 7	User provided, required.	The total memory for the computer, in Mbytes.
Operating System	ASCII characters	<= 60	User provided, required.	The operating system name/version for the computer.
Disks	ASCII characters	<= 60	N/A	<ul style="list-style-type: none"> • A list of the disks previously defined for that site. • This list of disks is used to select from when a disk is associated (or disassociated) with the computer using the arrow buttons.
Associated Disks	ASCII characters	<= 60	User provided, required.	Disks in this list are associated with the computer.
Comments	ASCII characters	unlimited	User provided, optional.	User comment on the resource.

4.7.1.3.4 String Details GUI

The String Details GUI, shown in Figure 4.7.1-12, is used to define the characteristics of a string, or a collection of virtual computer resources.



Figure 4.7.1-12. String Details GUIs

The characteristics of the other fields are described in Table 4.7.1-7.

Table 4.7.1-7. String GUI Field Description

Field Name	Data Type	Size (characters)	Entry	Description
Resource Name	ASCII characters	<= 60	User provided, required.	A user defined name for the resource.
Computers	ASCII characters	<= 60	N/A.	A list of the computers previously defined for that site. This list of computers is used to select from when a computer is associated (or disassociated) with the string using the arrow buttons.
Associated Computers	ASCII characters	<=60	User provided, required.	Computers in this list are associated with the string.
Comments	ASCII characters	unlimited	User provided, optional.	User comment on the resource.

Note that the Real Computer Details GUI and the Autosys Details GUI are similar to the String Details GUI. They are shown, respectively, in Figures 4.7.1-13 and 4.7.1-14.

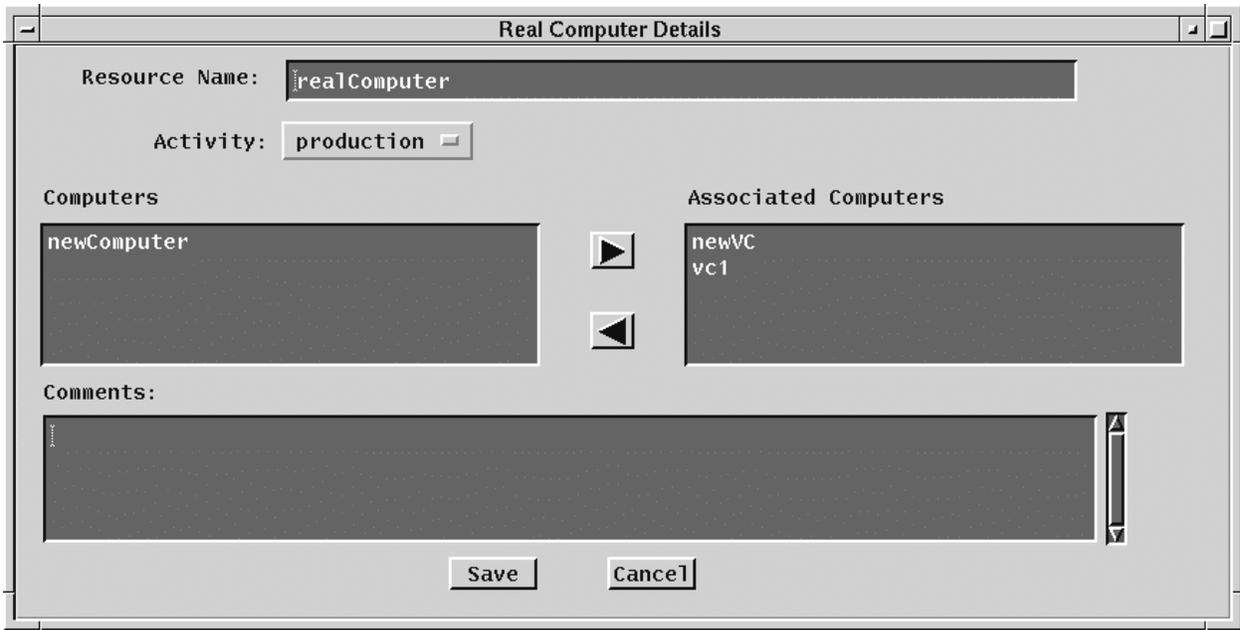


Figure 4.7.1-13. Real Computer Details GUIs

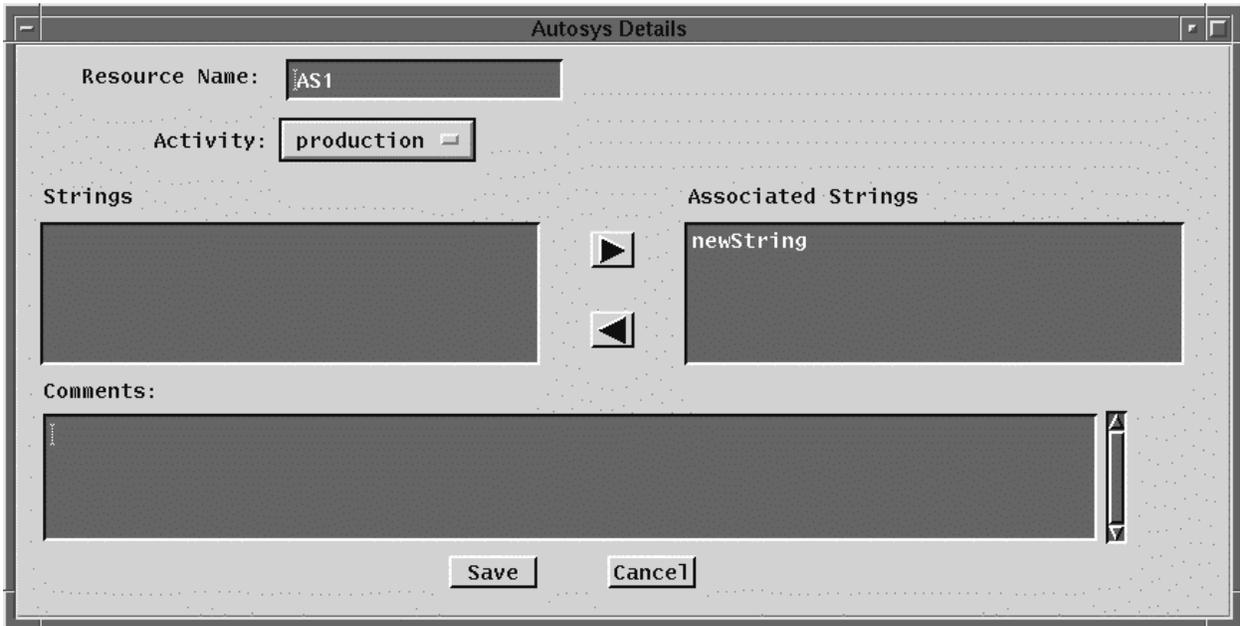


Figure 4.7.1-14. Autosys Details GUIs

4.7.1.4 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

4.7.1.5 Databases

The Resource Planning GUIs relies on the PDPS database to provide all the necessary information at the start time as well as to save all the updated input data about Resource Planning and Resource Definition.

Information on the PDPS database is contained within the Production Planning tools portion of this document, Section 4.8.1.3.

4.7.1.6 Special Constraints

There are no special constraints associated with the Resource Planning tools.

4.7.1.7 Outputs

The Resource Planning tools provide output in 3 forms:

- displays to the GUIs (described above)
- updates to the PDPS database for planning (described in Section 4.8.1.3)
- generates reports(described in Section 4.7.1.8).

4.7.1.8 Event and Error Messages

The resource planning tools provides informational messages or warnings for minor errors . Significant resource planning events or errors are logged to the MSS Event Log.

Both event and error messages are listed in Appendix D.

4.7.1.9 Reports

Note, this functionality will be made available in the future releases.

This page intentionally left blank.

4.8 Production Planning

This section describes the Production Planning GUIs used by DAAC operators.

The Production Planning GUIs are used by the production planner to define the science processing jobs to be run at a DAAC. The jobs are defined in terms of Production Requests (PRs). A PR is essentially an order for data to be produced by the data processing subsystem. A single PR may lead to several jobs being run over time, or to a single job producing a single set of data. PRs apply to the processing of new data (standard PRs, or standing orders) or the reprocessing of existing data (reprocessing PRs).

The Planning subsystem uses the PR and information on the science processing software (known as a Product Generation Executive, or PGE) to prepare a Data Processing Request (DPR). A DPR corresponds to a single processing job.

Planning subsystem GUIs are used to enter or modify PRs, review DPRs, and produce a selection of production planning related reports. The production planner uses the GUIs to create plans for data processing from the PRs defined in the Production Planning Subsystem. The GUIs are also used by the production planner to activate or cancel a plan.

The Production Planning GUIs are packaged into three applications: the Production Request Editor, the Production Planning Workbench, and the Production Strategies User Interface. The Production Request Editor and Production Planning Workbench are accessible through separate icons from the desktop

This page intentionally left blank.

4.8.1 Production Request Editor

The Production Request Editor allows the user to submit production requests, which produce data products. The Production Request Editor is used to perform the functions described in Table 4.8.1-1.

Table 4.8.1-1. Common ECS Operator Functions Performed with Production Request Editor

Operating Function	GUI	Description	When and Why to Use
Generate/Review /Edit Production Requests (PRs)	Production Request Editor – PR Edit, PR List	<ul style="list-style-type: none"> The operator enters Production Request information Production Requests define processing over a period of time 	To process or reprocess ECS data When a PR expires, a new one may need to be input
Review Data Processing Requests (DPRs)	Production Request Editor – DPR View, DPR List	<ul style="list-style-type: none"> The operator can review the characteristics of a particular data processing request DPRs are created automatically from PRs 	A DPR may be reviewed to inspect job parameters

4.8.1.1 Quick Start Using Production Request Editor

To execute the Production Request Editor from the command line prompt, enter:

```
>EcPIP_RE_IFStart <mode>
```

where:

<mode> is the ECS mode for the execution, e.g., **OPS**, **TS1**.

Refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series, for a listing of EcPIP_RE_IFStart.

4.8.1.2 Production Request Editor Main Screen

The Production Request Editor is a collection of display areas in a ‘tab stack’ selection arrangement. Clicking on a tab causes the associated tab “page” (also referred to simply as the tab) to be displayed. There are “pop-ups” associated with some of the tabs which expand the displayed area and provide GUI interaction for sub-functions. The first tab is ‘**Planning**’ shown in Figure 4.8.1-1. This is the ‘cover page’ or default of the tab stack, which lists and describes the other tabs. The primary activities associated with the Production Request Editor activity are:

- Production Request Editor Tab
- Production Request List Tab

- Data Processing Request View Tab
- Data Processing Request List Tab

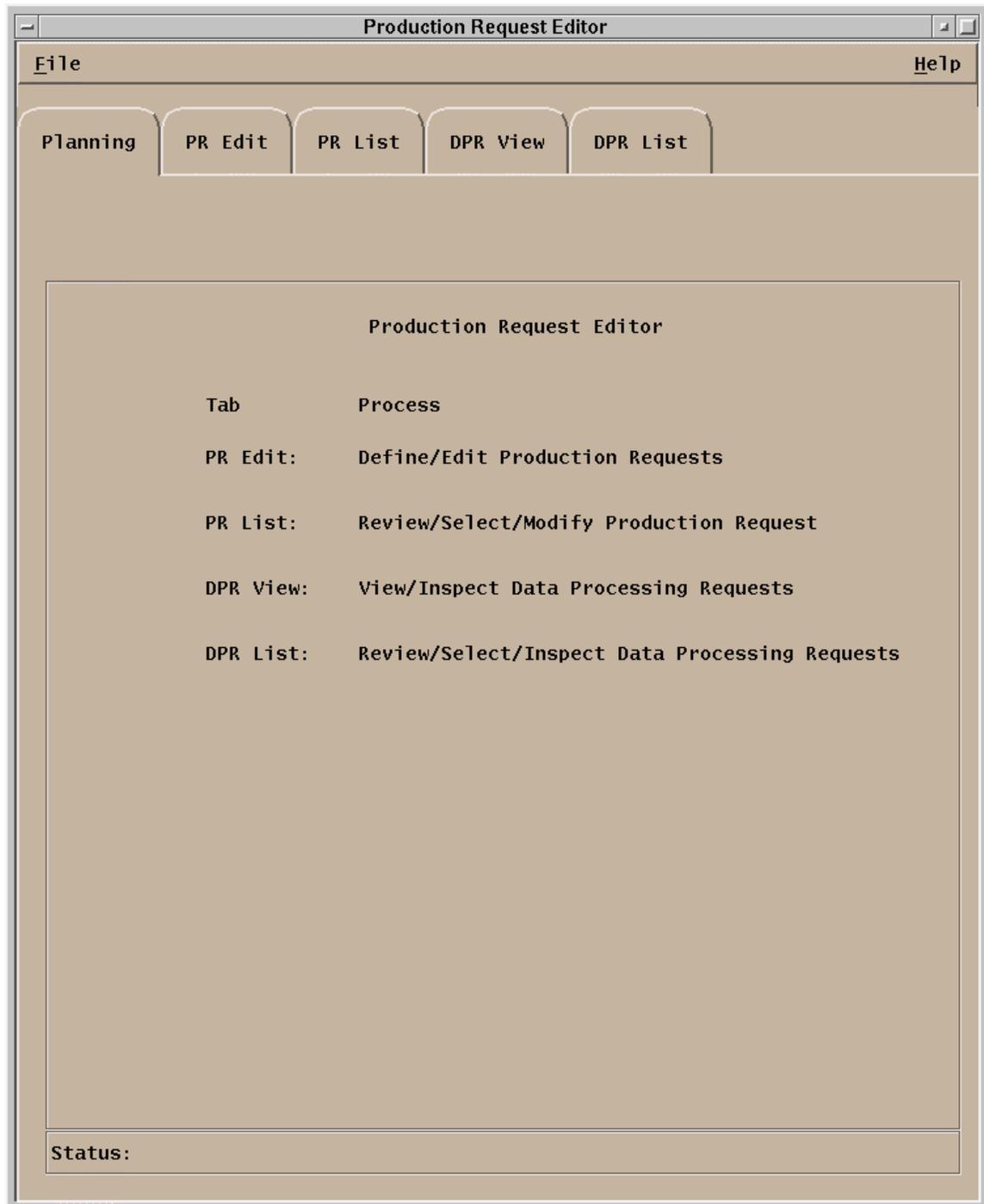


Figure 4.8.1-1. Production Request Editor GUI Showing the Planning Tab

In addition, on the menu bar, the pulldown menus provide the following capabilities.

- **File** Pulldown
- **Exit** – To exit the application.

4.8.1.2.1 PR List and PR Edit Tabs

The **PR List** tab, shown in Figure 4.8.1-2, and the **PR Edit** tab, shown in Figure 4.8.1-3, much commonality. The data displayed on the **PR List** tab is a subset of the data included on the **PR Edit** tab. These data fields are described in Table 4.8.1-2.



Figure 4.8.1-2. PR List Tab

The **PR List** tab allows the operator to review those PRs that have already been entered into the system. The PRs are presented in the order in which they appear in the PDPS database as a one line summary display for each PR. A particular PR can be located by entering a search string (including wildcard characters) in the field next to the 'Find' button and then clicking on the button. The first occurrence of the search string will be highlighted. By clicking on (highlighting) a PR and then selecting the **File: Open** pulldown option, the **PR Edit** tab is initiated for that selected PR.

The menu bar for the **PR List** tab and its pulldown menus provide the following capabilities.

- **'File'** Pulldown:
 - **Open** – Allows the operator to open a highlighted, existing Production Request for review or editing in the **PR Edit** tab.
 - **Exit** – To exit the application.
- **'Edit'** Pulldown:
 - **Delete** – To delete a production request.

The data on the **PR List** tab are identified by the column headings at the top of the display. These data descriptions are given in Table 4.8.1-2.

Production Request Editor

File Edit **Help**

PR Name:
 Origination Date: (UTC)

PR Type:
 Originator:

User Type:
 Priority:

Satellite Name:

Instrument Name:

PGE Name:

PGE Version:

Profile Id:

Collection Time
 Insertion Time

Duration UTC Time Orbit

Collection Time

Begin / / - : :

End / / - : :

Tile Id

From

To

Intermittent DPR

Skip
 Keep
 SkipFirst

Comment:

Figure 4.8.1-3. PR Edit Tab

On the menu bar for the **PR Edit** tab, the pulldown menus provide the following capabilities.

- **'File'** Pulldown:
 - **New** – When selected, the fields are cleared to allow entry of a new Production Request.
 - **Open** – Allows the operator to select an existing Production Request for review or editing. This function uses the **File Selection Pop-up** (Figure 4.8.1-4) invoked by **Save As**.
 - **Save As...** - Allows the operator to save the displayed Production Request and give it a new PR name. This function uses the **File Selection Pop-up** (Figure 4.8.1-4) invoked by **Open**.
 - **Exit** – To exit the application.
- **'Edit'** Pulldown:
 - **Delete** – To delete a production request.

Most PRs will be slight modifications of existing PRs, for example, to change the time duration of a PR. The process for entering a new PR is usually to select an available PR via the **File: Open** feature. The user then would modify the parameters as necessary and save the new PR through the **'Save As...'** option on the **'File'** menu bar of **PR Edit**. This action brings up a separate pop-up to name the new PR. Also, a completely new PR may be entered directly via the **PR Edit** tab.



Figure 4.8.1-4. File Selection Pop-up

The **PR Edit** tab fields are organized into six regions: Production Request Identification, Request Definition, Duration, Intermittent DPR, Comment, and Status. The individual fields of the **PR Edit** tab are described in Table 4.8.1-2.

Table 4.8.1-2. PR Edit Field Description (1 of 2)

Field Name	Data Type	Size (# of characters)	Entry	Description
Production Request Identification	--	--	--	Information used to identify the PR and the originator
PR Name	ASCII characters	<27	user input, required	a name for the PR
PR Type	GUI selection	N/A	click	Routine On Demand Reprocessing Ad-hoc Reprocessing
User Type	GUI selection	N/A	click	Operator DAAC Manager Scientist Researcher
Origination Date (UTC)	Date	10	system generated	date of PR entry
Originator	ASCII characters	<25	system generated	user ID of the user entering the PR.
Priority	Integer	<99	user input, required	priority to be associated with the PR; 1≤value≤99
Production Request Definition	--	--	--	Information defining the PR
Satellite Name	ASCII characters	<25	system generated	satellite name, if applicable, with which the PR/PGE is associated; displayed when PGE is selected.
Instrument Name	ASCII characters	<20	system generated	instrument name, if applicable, with which the PR/PGE is associated; displayed when PGE is selected.
PGE Name	ASCII characters	<12	system generated	name of PGE to be used in the PR; displayed when PGE is selected.
PGE Version	ASCII characters		system generated	The version number of the PGE to be associated with the PR; displayed when PGE is selected.
Profile ID	Integer	<99	system generated	The Profile Id of the PGE to be associated with the PR displayed when the PGE is selected.
CollectionTime				Data start/stop time
Insertion Time				Time ASTER tape was inserted by the DAAC
Duration	--	--	--	Time range over which the PR is applicable
UTC Time				
Orbit				

Table 4.8.1-2. PR Edit Field Description (2 of 2)

Field Name	Data Type	Size (# of characters)	Entry	Description
Start Date	Date	8	user input, required	start date of instrument data to be processed
Start Time	Time	6	user input, required	start time of instrument data to be processed
End Date	Date	8	user input, required	end date of instrument data to be processed
End Time	Time	6	user input, required	end time of instrument data to be processed
From (Orbit)	Integer	<8	user input	User can select orbit from number as an alternative to UTC Time.
To (Orbit)	Integer	<8	user input	User can select orbit to number as an alternative to UTC Time.
Tile Id	Integer	<8	User input	User enters the number of the tile requested.
Intermittent DPR				
Skip	Integer	<6	user input	Number of DPRs to skip
Keep	Integer	<6	user input	Number of DPRs to keep
SkipFirst	Flag	GUI	user selected	Determines whether the first DPR is skipped or kept.
Comment	ASCII characters	<255	user input, optional	user comment associated with the PR
Status	ASCII characters	<20	system generated	current status of the PR

There are 4 function selection buttons in the Request Definition section of the screen:

4.8.1.2.1.1 PGE Selection Button

As a part of the **PR Edit** process, a PGE will need to be associated with the PR. The PGE Selection pop-up contains a list of the PGEs available for selection and is accessed via the 'PGE...' button on the **PR Edit** tab. The **PGE Selection** pop-up is shown in Figure 4.8.1-5. The information describing each PGE is entered through GUIs used in the SSI&T process, described in Section 4.5.1. To associate a PGE to a PR, the desired PGE is highlighted from the list of PGEs, and then selected by clicking the **OK** button on the PGE Selection pop-up. A particular PGE may be searched for by entering in a search string (including standard wildcard characters) in the field next to the '**Find**' button and then clicking on the button. The resulting PGE list will match the search string in any of the displayed fields: PGE Name, PGE Version, etc. Clicking on the **Cancel** button at the bottom of the PGE Selection pop-up returns to the previous window.

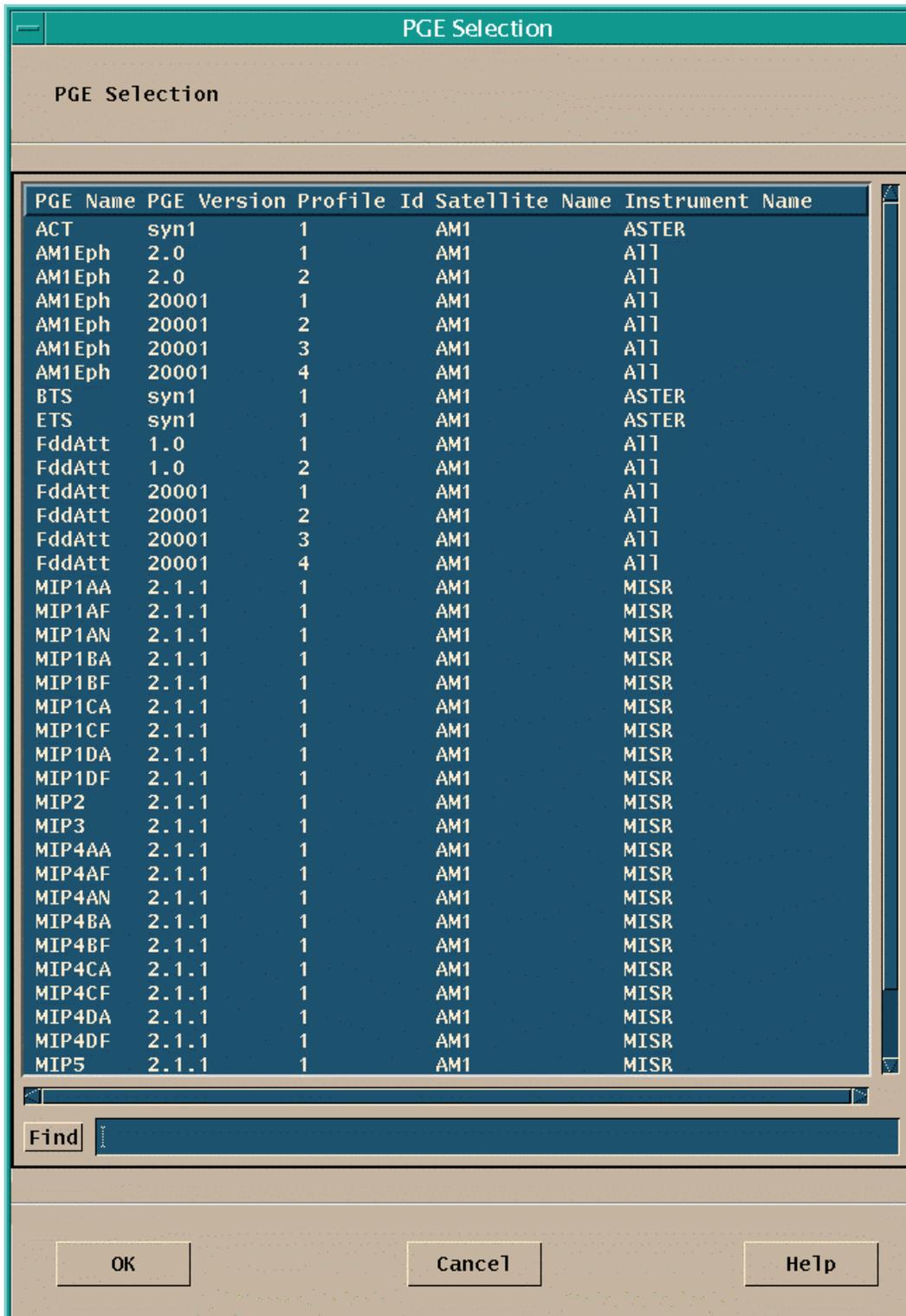


Figure 4.8.1-5. PGE Selection Pop-up

This screen has three function buttons:

- OK** Complete the action displayed (the selection)
- Cancel** Ignore the action displayed (the selection)
- Help** Display a Help pop-up with information about the function of this window.

The data displayed on this screen is described in Table 4.8.1-2.

4.8.1.2.1.2 PGE Parameters Button

The parameters that may be associated with a PGE when it is run are normally specified for the PGE when the PR is defined. If unspecified, the parameters default to values set during the SSI&T process, as described in Section 4.5.1. The production planner may edit or review these parameters by clicking the '**PGE Parameters...**' button on the **PR Edit** tab (Figure 4.8.1-3) once a PGE has been selected. The **PGE Parameter Mappings** pop-up that is used to edit/view the parameters are shown in Figure 4.8.1-6. The name of the PGE is shown in the text region at the top of the window. The window lists each parameter on a line with its default value. A particular PGE Parameter can be located by entering a search string (including standard wildcard characters) in the field next to the '**Find**' button and then clicking on the button. The resulting list of PGE Parameters will match the search string in any of the displayed fields: Parameter Name, Logical ID, etc. To override a value, select the desired parameter line, enter the override value in the 'Parameter Mapping' text region at the top, press enter and then click **OK**.

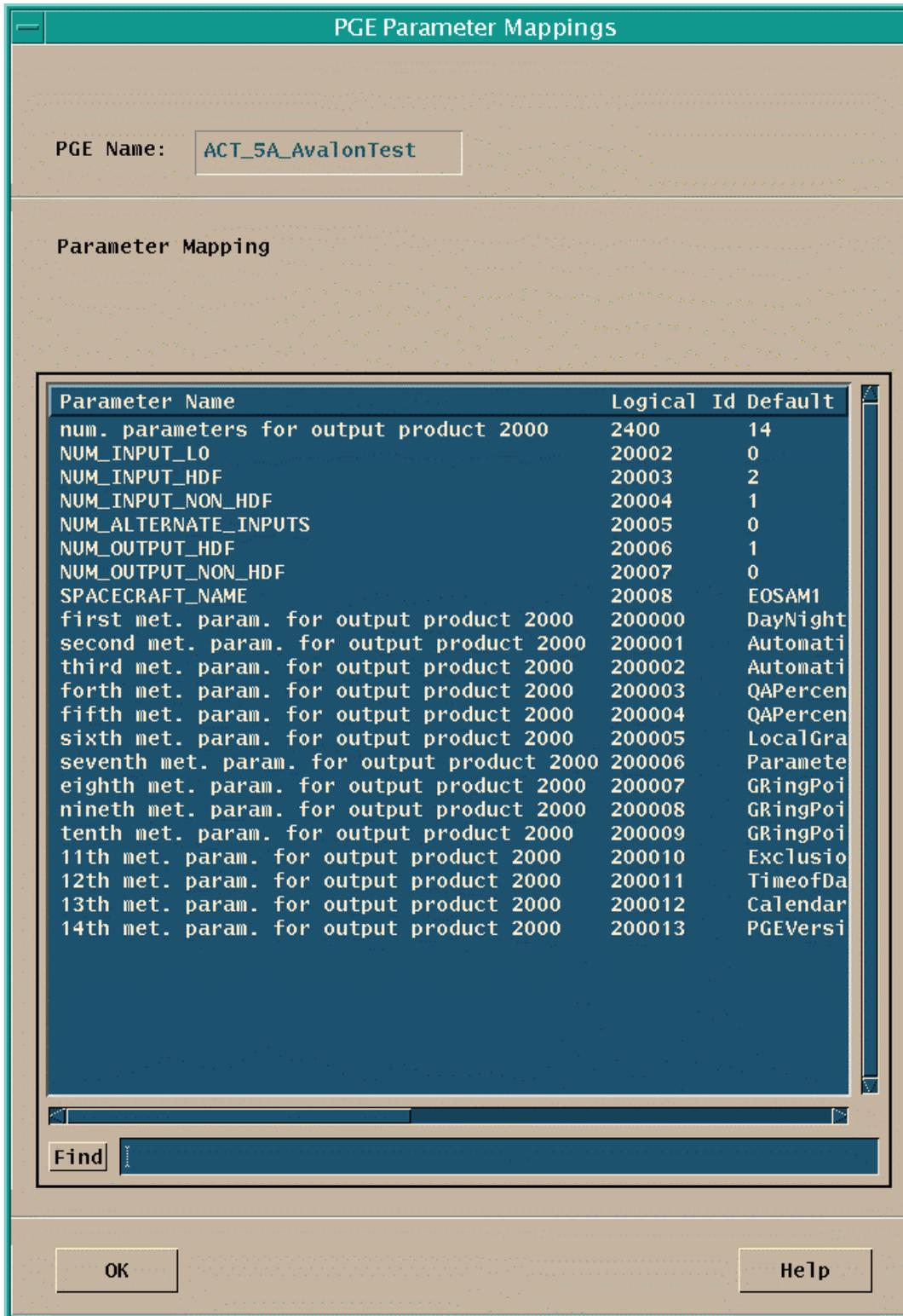


Figure 4.8.1-6. PGE Parameter Mappings Pop-up

This screen has three function buttons:

- **OK** Complete the action displayed (the selection)
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window.

The data displayed on this screen is described in Table 4.8.1-2.

4.8.1.2.1.3 Metadata Checks Button

The Metadata checks button on the PR Edit tab invokes the **MetaData Checks** pop-up window shown in Figure 4.8.1-7 to appear. This function allows the operator to check a metadata value against a specified input data granule metadata field value to determine if the data granule can be used or a PGE performed to update the values.

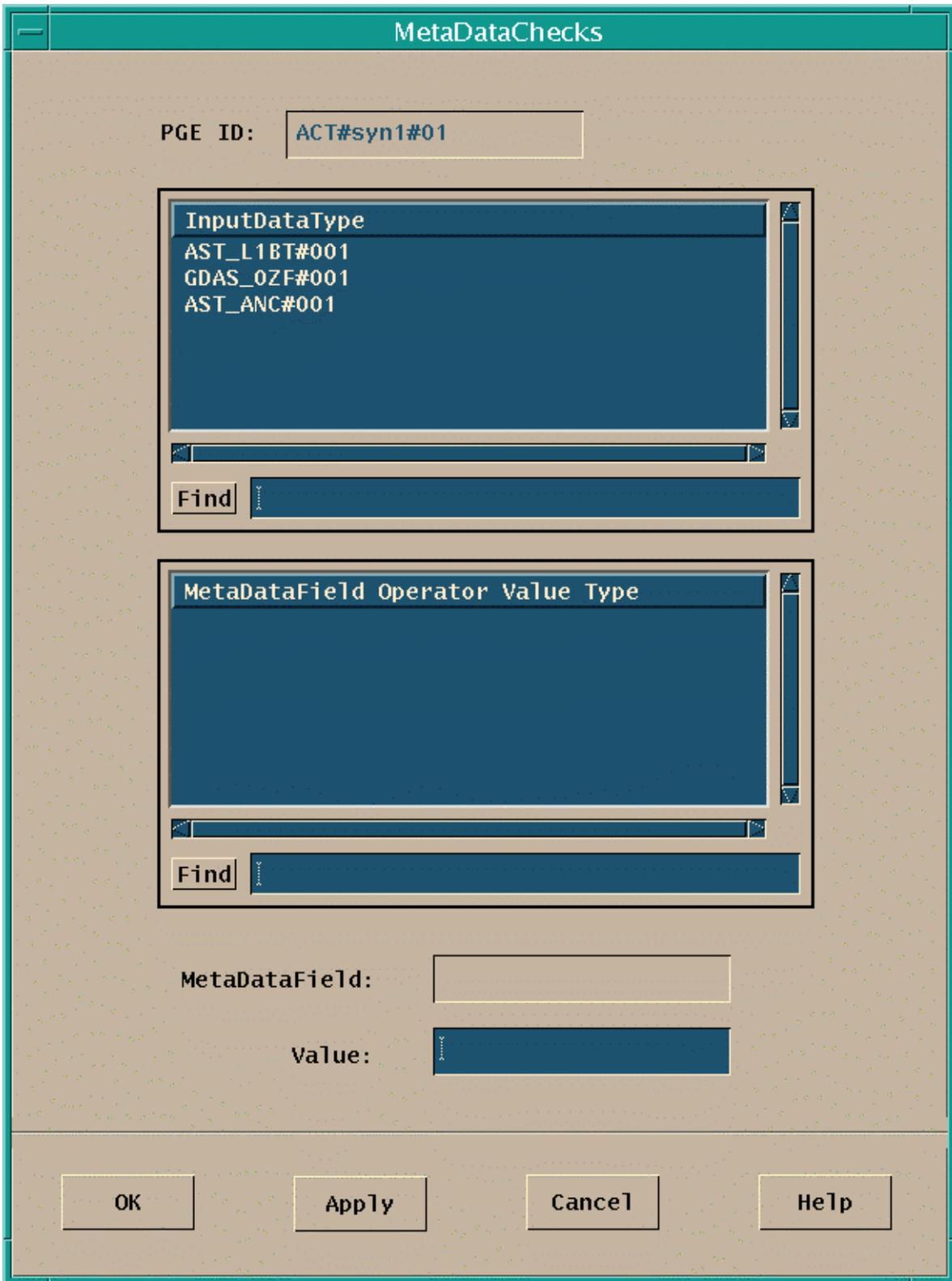


Figure 4.8.1-7. MetaData Checks Pop-up

This screen has four function buttons:

- **OK** Complete the action displayed (the selection)
- **Apply** Update the database with the value entered.
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window
-

Table 4.8.1-3 describes the fields displayed on the **MetaData Checks** pop-up window.

Table 4.8.1-3. PR Edit-MetaData Checks Field Description

Field Name	Data Type	Size (# of characters)	Entry	Description
PGE ID	ASCII characters	<17	system generated	ID of the associated PGE
Input Data Type	ASCII characters	<20	system generated	Identifier of the database containing the displayed metadata field value
MetaData Field	ASCII characters	<40	system generated	Name of the metadata field
Operator	ASCII characters	<3	system generated	The logical operator (<,>=, <=,+>) used with the metadata field.
Value	ASCII characters	<80	system generated	The value stored in the selected metadata field.
Type	ASCII characters	<5	system generated	Metadata field characteristic (Float, Int, String).

4.8.1.2.1.4 Alternate Input Values button

The Alternate Input Values button on the PR Edit tab causes the **AlternateInputValues** pop-up shown in Figure 4.8.1-8 to appear. This function allows the operator to select a replacement for a data input to a PGE with multiple inputs. This may be necessary if inputs to the PGE are not available and alternative inputs can be used.

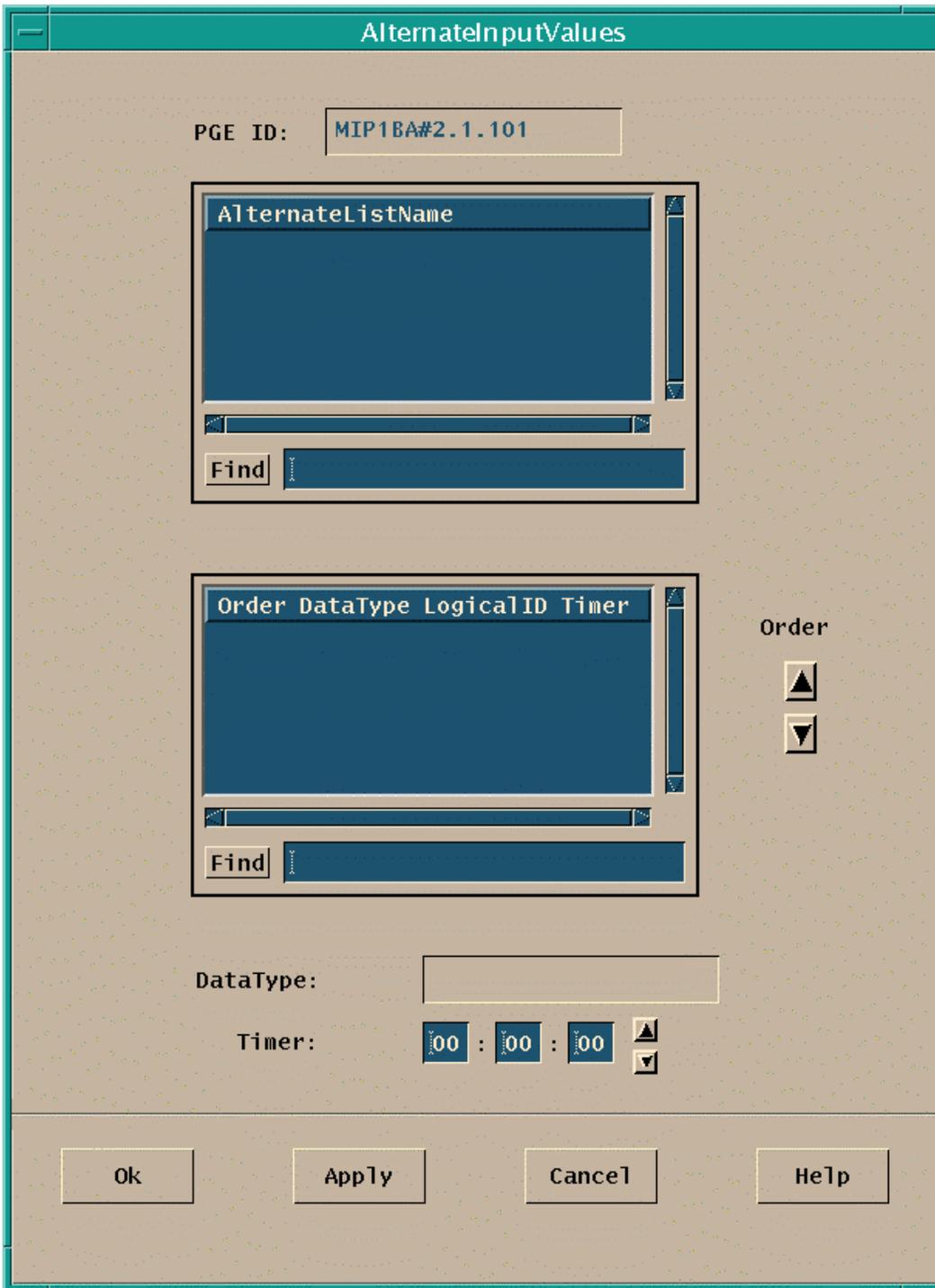


Figure 4.8.1-8. AlternatInputValues Pop-up

This screen has four function buttons:

- **OK** Complete the action displayed (the selection)
- **Apply** Update the database with the value entered.
- **Cancel** Ignore the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window

Table 4.8.1-4 describes the information displayed on the AlternatInputValues pop-up.

Table 4.8.1-4. PR Edit- AlternatInputValues Field Description

Field Name	Data Type	Size (# of characters)	Entry	Description
PGE ID	ASCII characters	<17	system generated	ID of the associated PGE
Alternate List Name	ASCII characters	<20	system generated	Identity of the primary data type for this input.
Order	Integer	<99	system generated/ User changeable	Current order of this data type
Data Type	ASCII characters	<20	system generated	Identity of the data type
Logical ID	Integer	<8	system generated	The SDP Toolkit logical identifier used to reference the data type
Timer	Long Integer	<12	system generated/User changeable	The time period the Subscription Manager will wait for an alternate input to arrive.

4.8.1.2.2 DPR View and DPR List Tabs

The **DPR List** tab, shown in Figure 4.8.1-9, and the **DPR View** tab, shown in Figure 4.8.1-10 have a lot of commonality. The data displayed on the **DPR List** tab is a subset of that included on the **DPR View** tab. Also, the dropdown menus of both tabs have the same options.

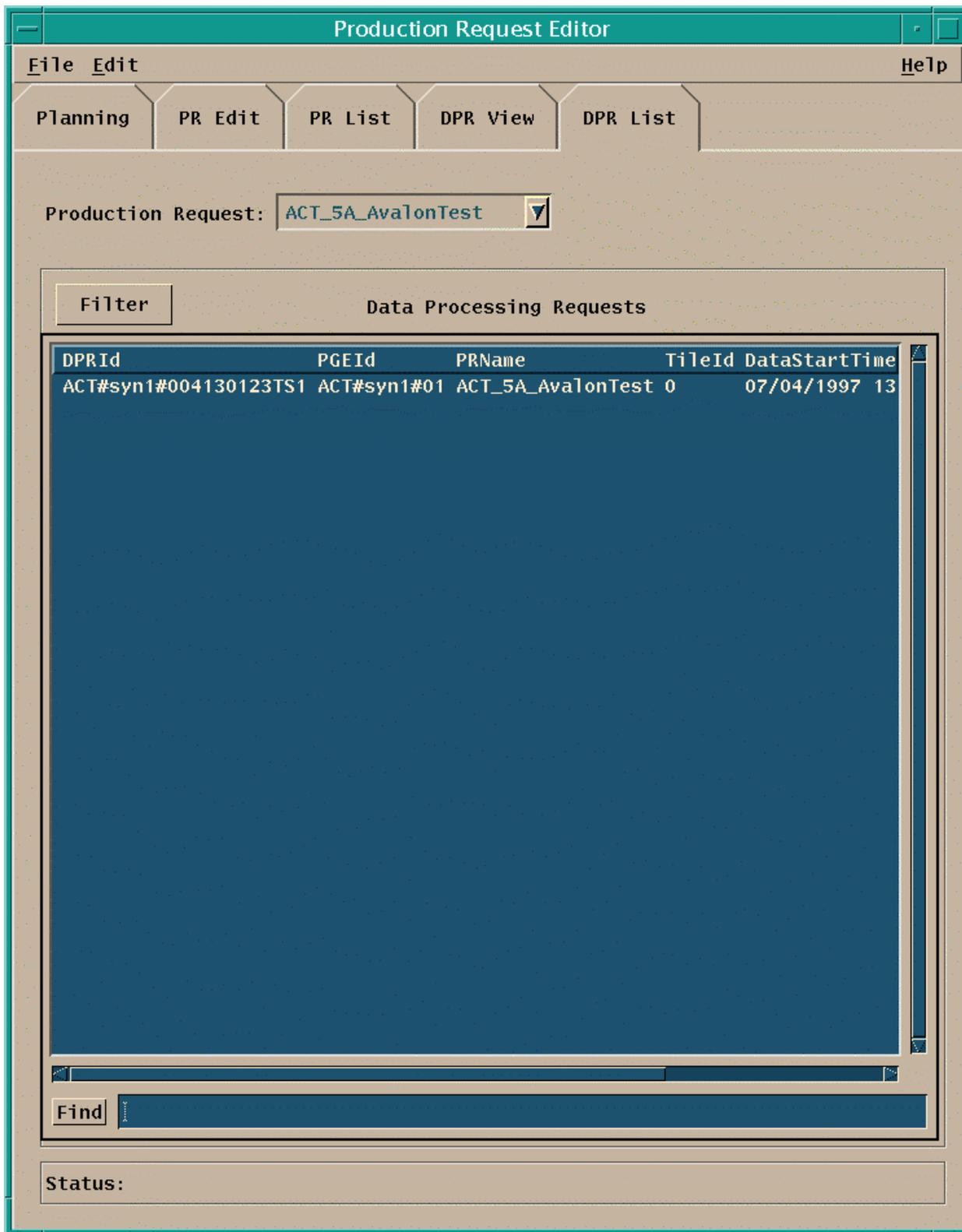


Figure 4.8.1-9. DPR List Tab

DPRs are generated automatically from the PRs described above during Planning. The Production Request Editor GUI provides capabilities to review these DPRs. By selecting the '**DPR List**' tab, a list of DPRs is displayed (in the order in which they have been entered into the PDPS database) for review in the same manner in which PRs are selected and edited. In addition, on the menu bar, the pulldown menus provide the following capabilities.

- '**File**' Pulldown:
 - **Open** - Allows the operator to select an existing DPR for review in the DPR View tab. In the **DPR View** tab (but not in the **DPR List** tab), this function uses the File Selection Popup (Figure 4.8.1-5).
 - **Exit** - To exit the application.
- '**Edit**' Pulldown:
 - **Delete** - To delete a DPR.

Each line of the **DPR List** display represents a DPR, i.e., a job that will be run when all data and resource needs are satisfied. A particular DPR may be searched for by entering in a search string (including standard wildcard characters) in the field next to the '**Find**' button and then clicking on the button. The resulting DPR list will match the search string in any of the displayed fields: DPR ID, PGE ID, etc. In addition, DPRs may be filtered for associated PRs by selecting a PR using the Production Request selection tool at the top of the window and clicking on the '**Filter**' button. By selecting (clicking on) one of the DPR summary lines and also selecting the '**DPR View**' tab, that DPR is displayed in detail..

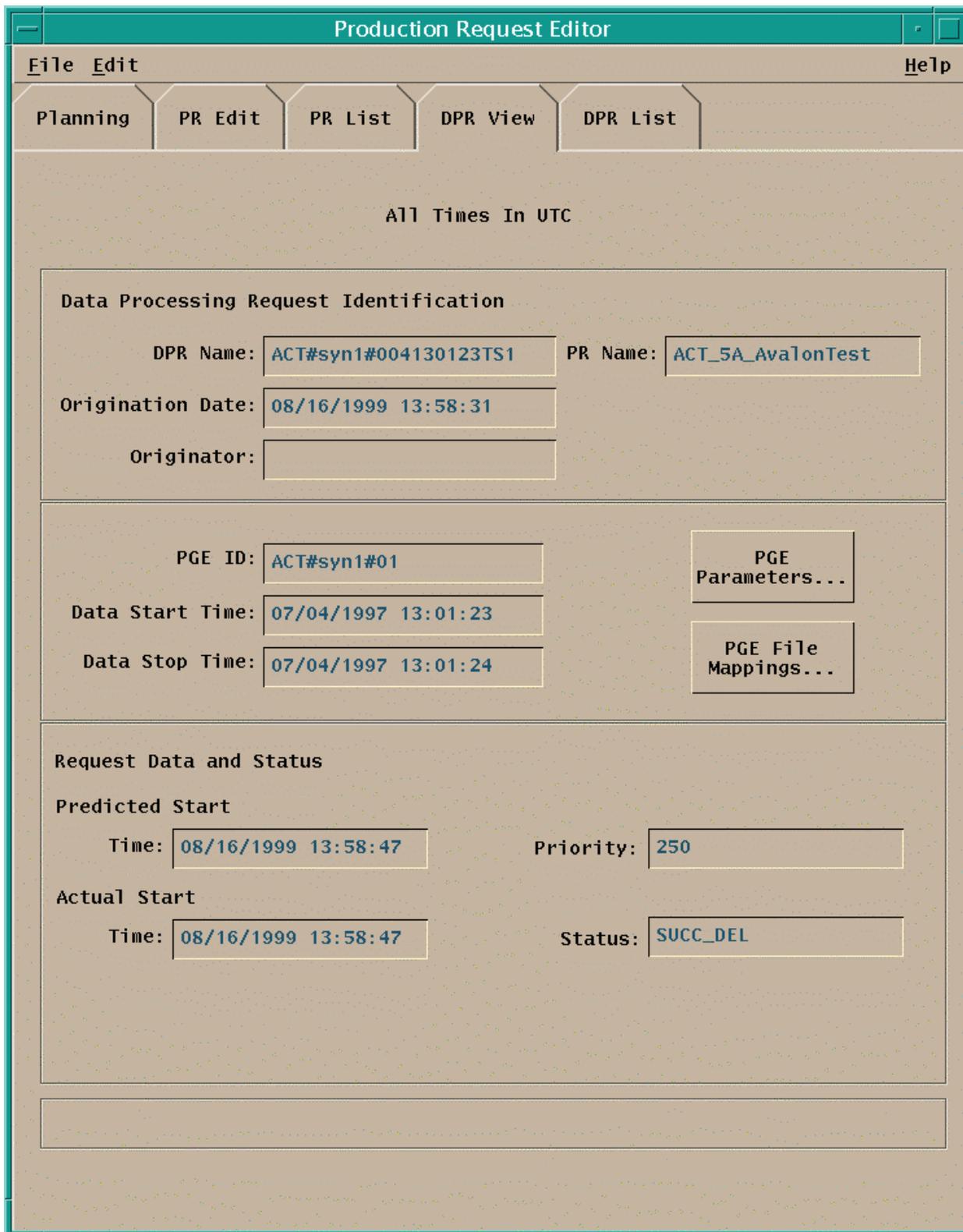


Figure 4.8.1-10. DPR View Tab

This screen has two function buttons:

- **PGE Parameters** See Figure 4.8.1-5
- **PGE File Mappings** See Figure 4.8.1-11

The **DPR View** tab fields are organized into three regions: Data Processing Request Identification, PGE Information, and Request Data and Status. The individual fields of the **DPR View** tab are described in Table 4.8.1-5.

Table 4.8.1-5. DPR View Field Description

Field Name	Data Type	Size (# of characters)	Entry	Description
Data Processing Request Identification	--	--	--	Information used to identify the DPR and the originator
DPR Name	ASCII characters	<24	system generated	DPR name generated from the associated PGE
PR Name	ASCII characters	<27	system generated	a name for the associated PR
Origination Date	Date	8	system generated	date of PR entry
Originator	ASCII characters	<25	system generated	user ID of the user entering the PR.
PGE Information	--	--	--	Information describing the PGE
PGE ID	ASCII characters	<17	system generated	ID of the associated PGE
Data Start Time	date & time	17	system generated	Start date and time of the data to be processed by the job
Data Stop Time	date & time	17	system generated	Stop date and time of the data to be processed by the job
Request Data and Status	--	--	--	Information describing the running status of the job
Predicted Start Time	date & time	17	system generated	Date and time at which the associated job is expected to be run, as predicted by the planning subsystem
Actual Start Time	date & time	17	system generated	Date and time at which the associated job ran
Priority	integer	3	system generated	priority of the job; 100≤; ≥1
Status	ASCII characters	<20	system generated	Status of the job

DPRs are generated automatically from PRs during the Planning process, using PGE information defined during SSI&T. The PGE information can be viewed but not edited from this screen. This

is because the information is quite complex and its alteration can have considerable side effects on the PGE execution.

PGE parameters may be viewed for a particular DPR run. These parameters may be accessed by selecting the '**PGE Parameters...**' button on the DPR View. This screen is similar to Figure 4.8.1-6 shown earlier.

The input and output files for a particular DPR can be viewed by selecting the '**PGE File Mappings...**' button. The associated GUI is shown in Figure 4.8.1-11. Fields displayed on this GUI are described in Table 4.8.1-6. The **PGE File Mappings** GUI displays one line of information for each file that may be used by or be produced by the PGE. A particular input or output file can be searched for by entering in a search string (including standard wildcard characters) in the field next to the '**Find**' button and then clicking on the button. The resulting file list will match the search string in any of the displayed fields. Information displayed includes:

- **Logical ID** The ID or tag used within the PGE to access the file.
- **Granule ID** The Universal Reference used to identify the file uniquely.
- **Start/Stop Time** The start or stop date and time for the data contained in this file.

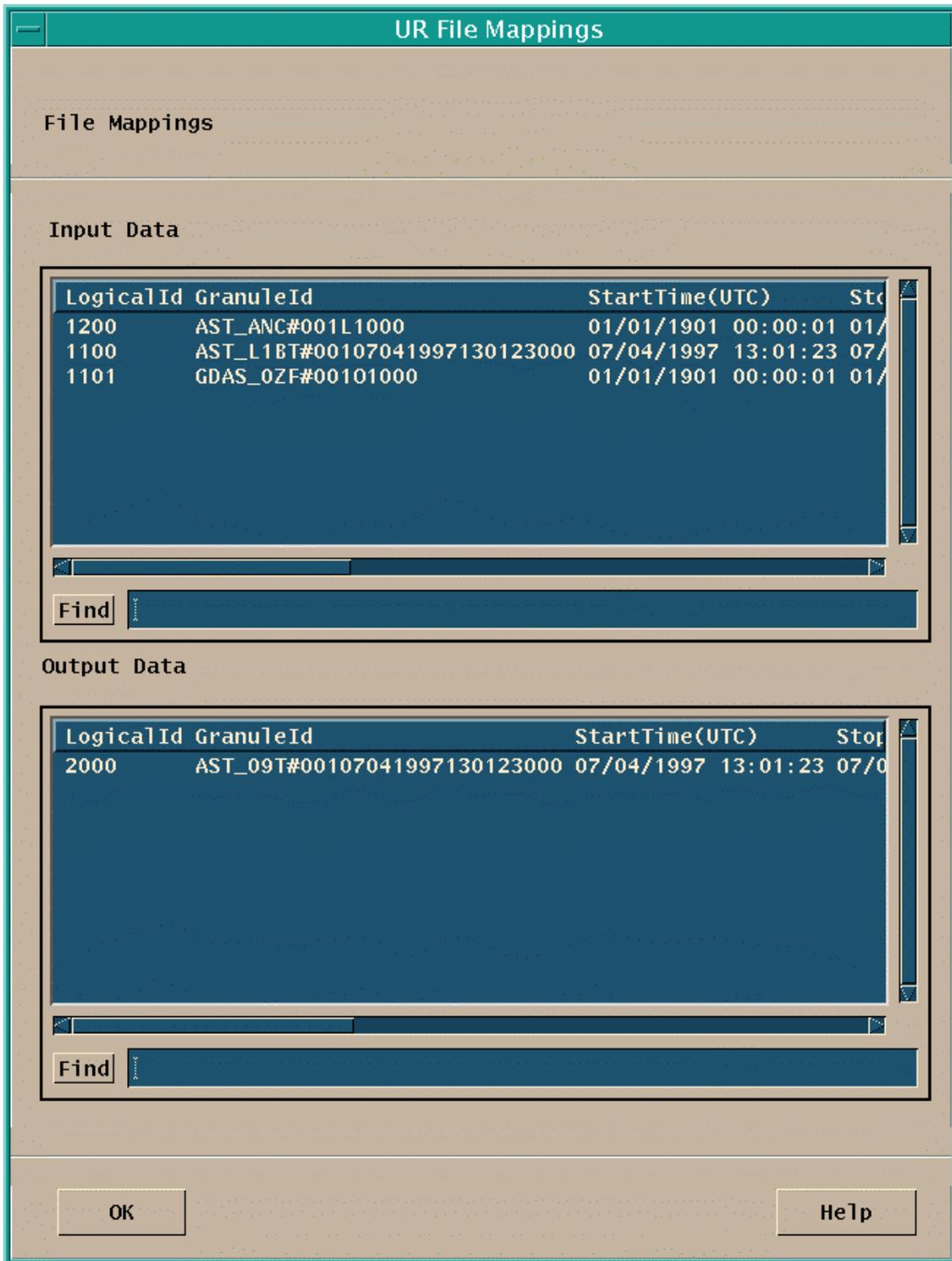


Figure 4.8.1-11. File Mappings Pop-up

This screen has two function buttons:

- **OK** Complete the action displayed (the selection)
- **Help** Display a Help pop-up with information about the function of this window.

Table 4.8.1-6. File Mappings Field Description

Field Name	Data Type	Size (# of characters)	Entry	Description
Logical ID	Integer	<8	system generated	The SDP Toolkit logical identifier used to reference the data type
Granule ID	ASCII characters	<20	system generated	Identity of the data type.
Start Time	Time	17	system generated	start time of instrument data
Stop Time	Time	17	system generated	end date of instrument data

4.8.1.3 Required Operating Environment

These GUIs may be hosted on the Planning Server and Queuing Server or Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

Table 4.8.1-7 identifies the supporting products this tool depends upon in order to function properly.

Table 4.8.1-7. Support products for Production Request Editor

Product
Sun Solaris 2.5
MOTIF 1.2.3
PDPS Database

4.8.1.3.1 Interfaces and Data Types

The Planning Subsystem has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information from the SSI&T components after the integration of a new or updated PGE. An element of the Planning system which does not require an operator interface, the Subscription Manager, also interfaces with the ECS Science Data Server Subsystem.

The Planning Subsystem is primarily intended for use by the operations staff of the EOSDIS DAACs. However, the capability has been requested and made available for science personnel from the SCFs to enter Production Requests into the Planning subsystem using the same interface as is described above. This is to be done with the consent of the DAAC operations managers and via the X-11 interface with procedures and equipment provided to ensure a secure interface for these updates.

4.8.1.4 Databases

The Planning Subsystem includes the PDPS database which contains information needed to plan the processing at a DAAC. It is also used for resource planning, containing information on the configured resources and their allocation. Finally, the database is used by the Data Processing Subsystem, the other major element of the PDPS. The following subsections provide a summary description of the PDPS database tables, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software, PGEs, within the production processing environment. As a part of this activity, the SSI&T GUI, described in Section 4.5.1 of this document, provide for the entry into the databases used by SSI&T of PGE-related information needed to plan for and run science software. At the conclusion of the SSI&T process, this database information is transferred to the operational databases from the databases established for SSI&T.

The Production Request Editor allows changes to scheduling information for PRs and DPRs. The Report Generator provides reports covering all the pertinent information in the PDPS database. The complete database schema is listed in 311-CD-106-005, *Planning and Data Processing Subsystem Database Design and Schema Specifications*.

4.8.1.5 Special Constraints

There are no special constraints that are associated with the Production Request Editor.

4.8.1.6 Outputs

Outputs of the Production Request Editor are provided in one of three ways:

1. Production Request Editor GUI responses as described above, including the File Mapping GUI.
2. Updates to the PDPS database (described in Section 4.8.1.3).

4.8.1.7 Event and Error Messages

The Production Planning GUIs provide informational messages or warnings for minor errors which the operator can immediately correct directly via the operator interface. Significant production planning events or errors are logged to the ECS Planning Server ALOG file. The Error messages are listed in Appendix A.

4.8.1.8 Reports

None.

This page intentionally left blank.

4.8.2 Production Planning Workbench

The second major element of production planning is the Production Planning Workbench (PWB). Table 4.8.2-1 summarizes the operational functionality of this tool. Via the GUI, the operator selects a Production Request (PR) to be included in the plan and then generates the DPR information needed to execute the PR. The planning tool provides a forecast of the start and completion times of the jobs based upon historical experience in running these PGEs. Using the planning tool, the operator can “activate” the plan, i.e., transfer the information included in the plan to the Data Processing subsystem and load it into the AutoSys tool where production processing is managed.

Table 4.8.2-1. Common ECS Operator Functions Performed with Production Planning Workbench

Operating Function	GUI	Description	When and Why to Use
Create a Plan	Production Planning Workbench	The operator can select available PRs to be included in the plan	When a new or revised plan needs to be prepared
Plan Activation	Production Planning Workbench	The operator can activate a plan, which transfers the planned jobs into the Data Processing System	When a new plan is prepared and accepted
Save Candidate	PWB	Create What-if scenarios	When plans are in the process of being created

4.8.2.1 Quick Start Using Production Planning Workbench

To execute Production Planning Workbench from the command line prompt, enter:

>EcPIAllStart <mode> <Application ID>

<mode> is the ECS mode under which the program is to run, e.g., OPS, TS1.

<Application ID> is an integer from 1 to 5 to uniquely identifying the execution.

4.8.2.2 Production Planning Workbench Main Screen

When the Production Planning Workbench is started, the GUI window shown in Figure 4.8.2-1 is displayed. Table 4.8.2-1 contains a description of the fields in this GUI. The GUI is started using information from the currently activated plan. Other plans may be opened or new plans created and saved using the ‘New’, ‘Open’, ‘Save’, and ‘Save As’ options on the ‘File’ pull down on the menu bar, much like creating, opening, and saving a text document.

Comments to the plan may be made in the indicated region. Selected PRs can be moved from one list to the other using the arrow buttons. A PR is scheduled by selecting from the pool of available ‘Unscheduled’ PRs and, using the arrow buttons, the selected PRs are moved to the ‘Scheduled’

list. A note of clarification about Ground Events during scheduling: All Ground Events will be automatically scheduled with any plan. Hence the GUI Ground Events will always be put into the 'Scheduled' list. Whenever a plan is activated, the Ground Events will also be activated. (If there is an unscheduled Ground Event in the 'Unscheduled' list, this means that this Ground Event has lost allocations.) The new plan can then be saved with a new name through the 'Save As' option on the 'File' pull-down menu.

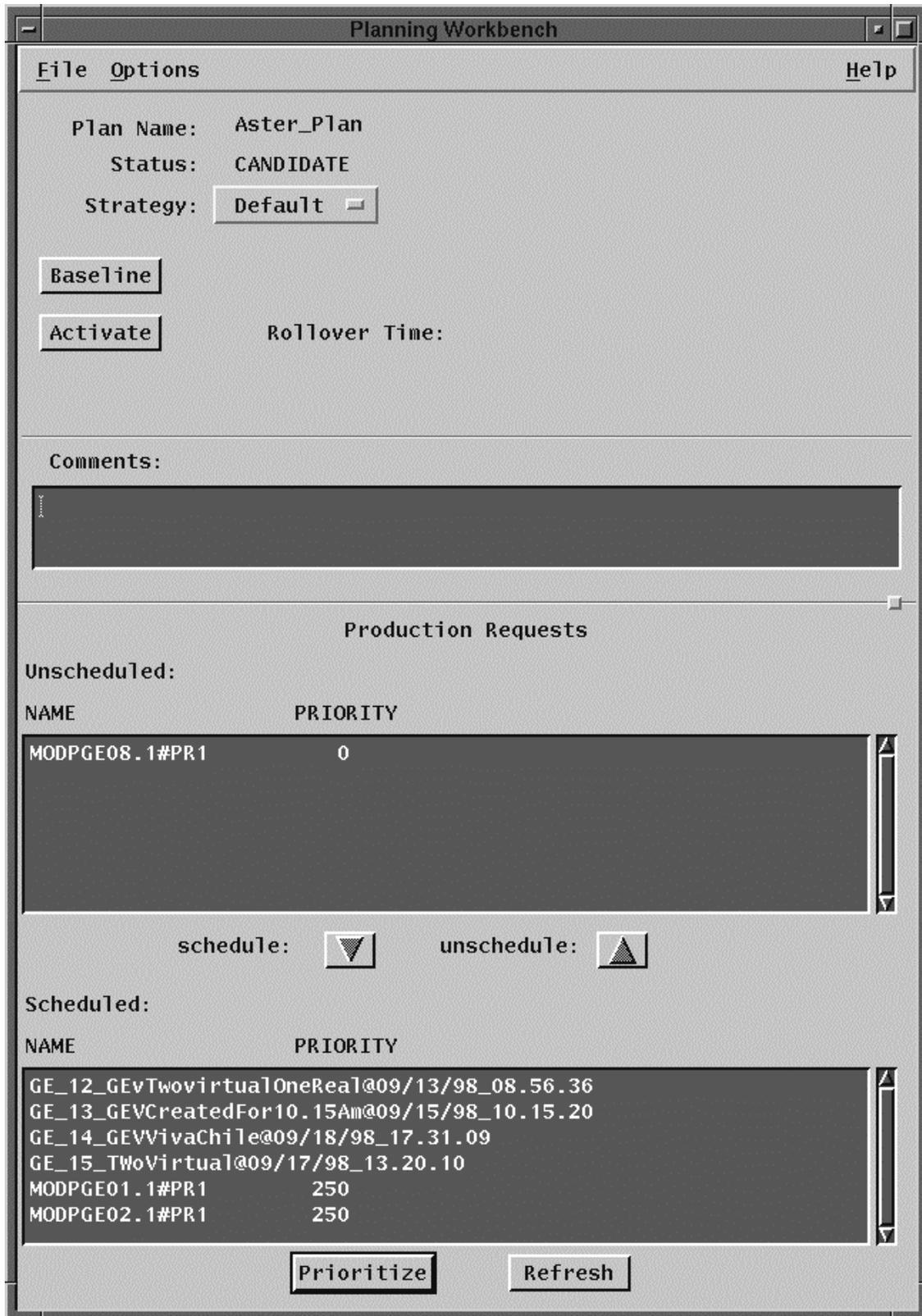


Figure 4.8.2-1. Production Planning Workbench GUI

Table 4.8.2-2 summarizes the information fields on the Production Planning Workbench GUI. Production Planning Workbench Pulldown Menu Options:

File

New clears all the fields, and creates a new plan. The new plan name can not be longer than 20 characters.

Open allows the user to select an existing Production Request for review or editing. This function uses the File Selection Pop-up (Figure 4.8.1-4) invoked by Save As

Save saves the current plan

Save As saves the displayed Production Request and gives it a new PR name. This function uses the File Selection Pop-up (Figure 4.8.1-4) invoked by **Open**.

Delete Allows user to delete a plan via a file selection pop-up. The active plan cannot be deleted but other candidate plans can be removed

Exit exits the application.

Options (no functionality has been associated with this menu, as yet).

The following push button is also available to the operator:

Baseline records the plan and the time of baselining. This baseline plan can be used as a point of comparison with which to compare future plans and results.

Activate activates a plan and the Data Processing Requests (DPRs) associated with the planned PRs are then transferred to the Data Processing subsystem and loaded into the AutoSys production queuing system. Once the necessary data dependencies are satisfied, the jobs are 'Released' to be run when processing resources are available. When the Activate button is pressed on the planning workbench, the current active plan get "replanned over" by the selected plan. PRs that are in the "old" active plan that are not selected in the "new".

Schedule schedule the selected PRs in the Unschedule List.

Unschedule un-schedule the selected PRs in the Schedule List.

Prioritize not yet available.

Refresh not yet available.

Also a pulldown list is available:

Strategy: selects different strategies for scheduling, which determine the priorities assigned to the individual PRs in a plan.

For all of these actions, the operator will be prompted for confirmation before proceeding.

Table 4.8.2-2. Production Planning Workbench Field Description

Field Name	Data Type	Size	Entry	Description
Plan Name	ASCII characters	<20	system generated	Name assigned to the plan; assigned initially when the 'New' plan is saved
Status	ASCII characters	<20	system generated	Status of the displayed plan: 'Active', 'Candidate'
Rollover Time	ASCII characters	19	system generated	The time at which the currently selected plan was activated.
Comment	ASCII characters	<256	user input, optional	User comments
Production Requests: Unscheduled	ASCII characters	<256	system generated	List of available PRs which are currently not scheduled
Production Requests: Scheduled	ASCII characters	<256	user input	List of available PRs which are currently scheduled Planner selects these from 'Unscheduled' list and moves them to/ from 'Scheduled' with arrow buttons

4.8.2.2.1 Planning Master Timeline GUI

A graphic, timeline-oriented depiction of the plan, as shown in Figure 4.8.2-2, can then be displayed. When this object is selected, the **Planning Master Timeline GUI** is displayed. This display is similar to the resource planning timeline display discussed in Section 4.7 of this document.

The **Planning Master Timeline GUI** represents a set of computers, arranged along the left side of the GUI, and some period of time as indicated across the top edge of the GUI. The execution of Data Processing Requests (DPR)s on a computer over a period of time is represented by several DPR bars across the GUI for that computer. A bar represents a time period during which a PGE (as described by a DPR) is running. Each bar has the name of the PGE. Scroll bars allow scrolling up and down through the full list of computers and left and right in time. A select list of time span viewing options (e.g., 24-hours, 48-hours) at the lower left of the screen is available for selecting the time span of interest. If one exits from the **Planning Master Timeline GUI**, it can be restarted as explained in Section 4.8.2.1.

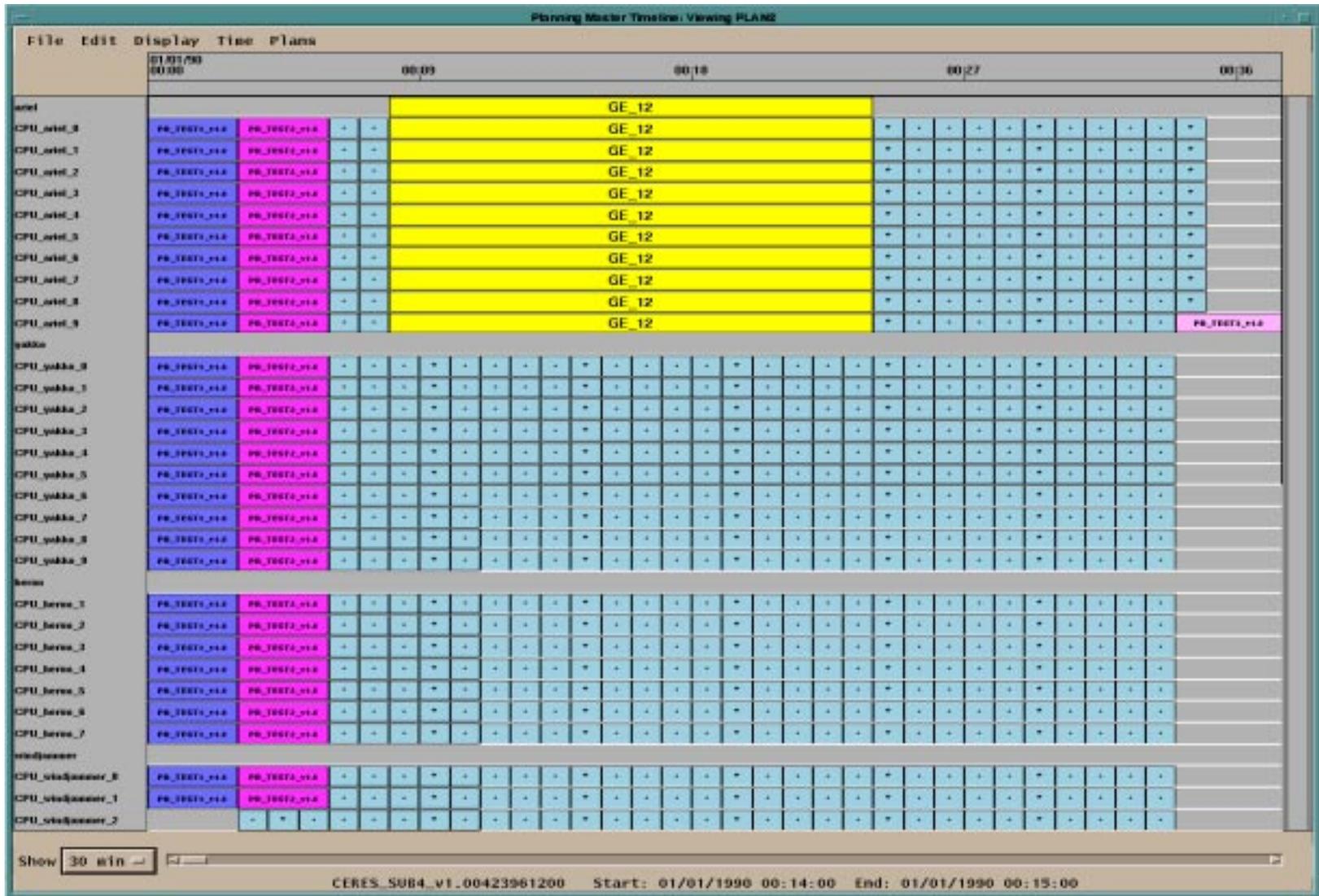


Figure 4.8.2-2. Planning Master Timeline GUI

The Planning Master Timeline Pulldown Menu Options are:

File

- **Open Plan:** Opens a previously created plan through a file selection pop-up. It loads it in the main region of the timeline and adds it to the plans menu buttons.
- **Load Configuration:** loads a configuration for the timeline containing a list of resources and color options. It does this through a file selection.
- **Save Configuration:** saves a configuration file.
- **Exit:** quits application.

Edit - Is not available in Release 4

Display - To select resources and attributes of the display: e.g. colors.

Time - To select Start and Stop times of the plan windows.

Plans - To elect plans to include

4.8.2.3 Required Operating Environment

These GUIs may be hosted on the Planning Server and Queuing Server, Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

Table 4.8.2-3 identifies the supporting products this tool depends upon in order to function properly.

Table 4.8.2-3. Support Products for Production Planning Workbench

Product
Sun Solaris 2.5.5
MOTIF 1.2.3
PDPS Database

4.8.2.3.1 Interfaces and Data Types

The Production Planning Workbench has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information from the SSI&T components after the integration of a new or updated PGE. An element of the Planning system which does not require an operator interface, the Subscription Manager, also interfaces with the ECS Science Data Server Subsystem.

4.8.2.4 Databases

The Production Planning Workbench uses the PDPS database which contains information needed to plan the processing at a DAAC. It is also used for resource planning, containing information on the configured resources and their allocation. Finally, the database is used by the Data Processing

Subsystem, the other major element of the PDPS. The following subsections provide a summary description of the PDPS database table, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software, PGEs, within the production processing environment. As a part of this activity, the SSI&T GUIs, described in Section 4.5 of this document, provide for the entry into the databases used by SSI&T of PGE-related information needed to plan for and run science software. At the conclusion of the SSI&T process, this database information is transferred to the operational databases from the databases established for SSI&T.

The complete database schema is listed in 311-CD-106-005, *Planning and Data Processing Subsystem Database Design and Schema Specifications*.

4.8.2.5 Special Constraints

There are no special constraints that are associated with the production planning tools.

4.8.2.6 Outputs

Outputs of the Production Planning Workbench are provided in one of three ways:

Production Planning Workbench GUI responses as described above, including the Production Plan Timeline display, Figure 4.8.2-3.

Updates to the PDPS database (see Section 4.8.2.4).

4.8.2.7 Event and Error Messages

The Production Planning Workbench provides informational messages or warnings for minor errors which the operator can immediately correct directly on the operator interface. Significant production planning events or errors are logged to the ECS Production Server ALOG file. Both event and error messages are listed in Appendix A.

4.8.2.8 Reports

Report generation capability is not available in Release 4.

4.8.3 Production Strategies User Interface

The Production Strategies User Interface is used to tailor the priority of production requests (PRs). Table 4.8.3-1 summarizes the operation functionality of this tool. The priority of production requests impacts the development of processing plans. Production Strategy is a method of developing a composite priority from several factors for a production request. These factors may be assigned different proportions of the final priority. The factors themselves are composed of attributes which may be given separate priorities. The factor attributes are then averaged to produce the priority of the factors.

The Production Strategies User Interface provides an option to display the default values for the factors and components.

Table 4.8.3-1. Common ECS Operator Functions Performed with Production Planning GUIs

Operating Function	GUI	Description	When and Why to Use
Manage Production Strategies	Production Strategies User Interface	The operator can view, create, modify, or delete Production Strategies.	When the priority of a PR needs to change to adjust processing scheduling.

4.8.3.1 Quick Start Using Production Strategies User Interface

To execute the Production Strategies User Interface from the command line prompt, enter:

```
>EcPIProdStratStart <mode>
```

<mode> is the ECS mode, e.g., OPS, TS1.

Refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series, for a listing of EcPIProdStratStart

4.8.3.2 Production Strategies User Interface Main Screen

The **Production Strategies** main screen is shown in Figure 4.8.3-1.

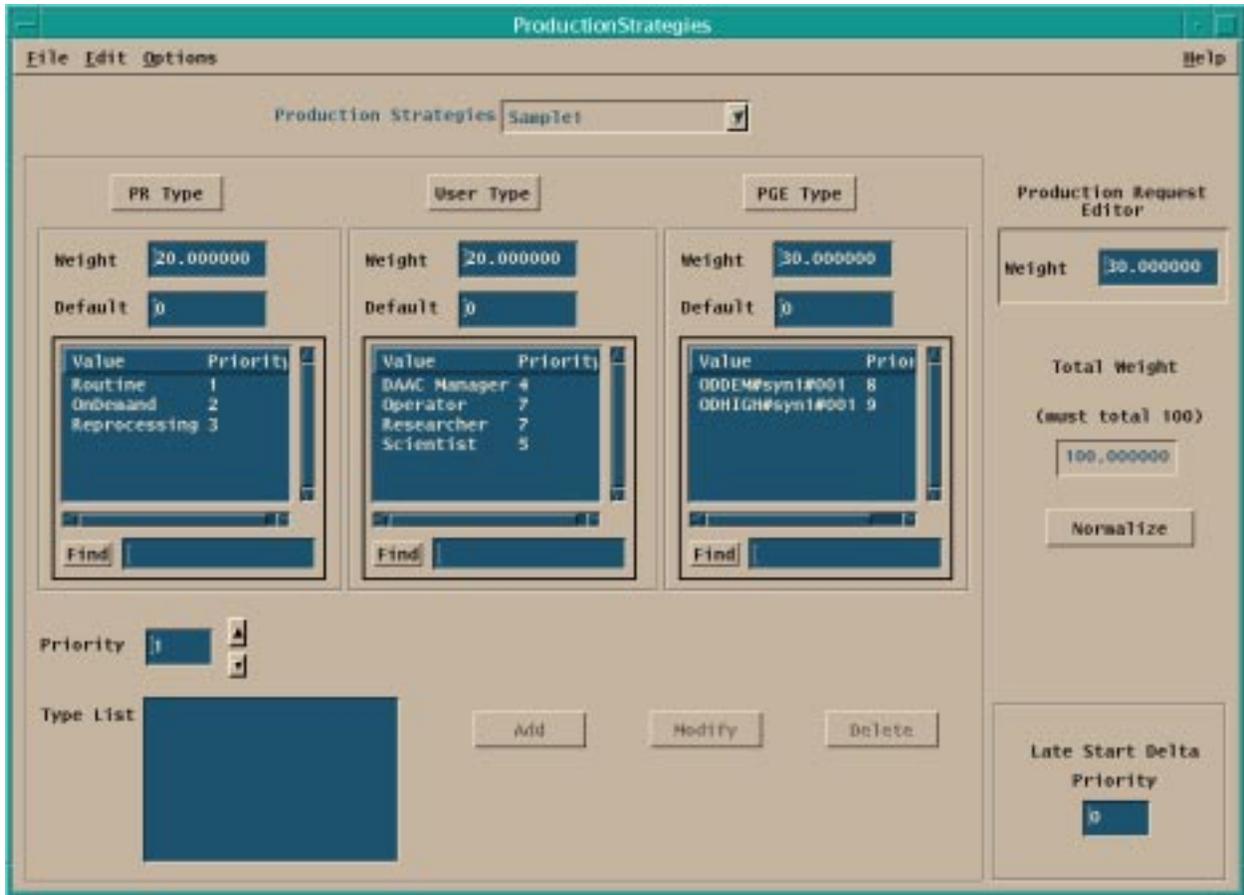


Figure 4.8.3-1. Production Strategies Main Screen

The menu bar for the Production Strategies User Interface GUI contains pulldown menus providing the following capabilities.

File Pulldown:

Open displays the Open Production Strategies pop-up window shown in Figure 4.8.3-2 with the production strategies currently in the PDPS database.

New clears all the display areas on the screen for input.

Save saves the displayed values to the database for the production strategy displayed.

Save As saves the displayed production strategy to the database as a different strategy.

Exit to exit the application.

Edit Pulldown:

Delete - To delete a production strategy.

Options Pulldown:

ActiveStrategy - Displays the **Active Production Strategy** screen shown in Figure 4.8.3-3).

Other options available on the Production Strategies screen:

Pr Type selects the PR Type panel as active.

User Type selects the User Type panel as active.

PGE Type selects the PGE Type panel as active.

Normalize adjusts the weighting values to total 100.

Add the Type List/Priority pairs displayed on the screen to the selected type.

Modify the Type List/Priority pairs displayed on the screen in the selected type.

Delete the Type List/Priority pairs displayed on the screen from the selected type.

The parameters associated with the Production Strategies User Interface are used as attributes and weights to generate the priorities for use by the overall Production Plan. All the parameters for the Production Strategies User Interface tools are accessible through the operator GUIs. These parameters are listed in Table 4.8.3-2. The parameters correspond to the sub-schema elements in the PDPS database referenced in Section 4.8.3.4 below.

The Production Strategies creates a two-level scheme for prioritizing the Production Plans. One level divides the plan into components: PR Type, User Type, PGE Type, and the Production Request Editor. Each of these components (except Production Request Editor) is further broken down into elements related to the Type. In the picture of the screen these elements may be observed in the list boxes under the type labels. Each of the elements may be assigned a priority.

The element priority assignment is initiated by selecting one of the type buttons. A type button brings one of the type frames into focus and displays the type elements in the Type List in the lower left corner of the screen. Element priorities are assigned or changed by selecting an element in the Type List, choosing a priority in the scroll box above the Type List, and clicking the appropriate button for Add, Modify, or Delete. The result will appear in the type frame list box with the value (element) and priority displayed. The Production Strategies GUI will re-compute the overall type priority based on the current element priorities of that type.

The second level of the priority scheme involves giving “weight” to the types. This “weight” is similar in concept to percentage. Note: the sum of the four “weight” values must equal 100. Each of the types may be given a weight. The fourth weight is the one assigned by the user in the Production Request Editor. As the values are entered, the program will total the weights and display an error dialog if the total is over 100.00. The program will not allow the entry of a weight that causes the total to exceed 100.00. If the total is less than 100.00 the operator may use the “Normalize” button to scale the weight values to base 100.00.

Table 4.8.3-2. Production Strategies Field Descriptions (1 of 2)

Field Name on screen	Data Type	Size (# of characters)	Entry	Description
Production Strategies	ASCII text--	20	Default = Default	Production Strategy Id
PR Type	--	--	--	Strategy component name (Production Request)
Weight	Floating Pt number	<100.	Default = 50.00000	Weight for this component in final priority computation
Default	Integer	<10	Default = 5	Priority for component.
Routine	Integer	<10	Default = 5	(Sub) priority for Routine Requests
OnDemand	Integer	<10	Default = 7	(Sub) priority for On-Demand Requests
Reprocessing	Integer	<10	Default = 3	(Sub) priority for Reprocessing Requests
User Type	--	--	--	Strategy component name (User Type – all users)
Weight	Floating Pt number	<100.	Default = 00.00000	Weight for this component in final priority computation
Default	Integer	<10	Default = 5	Priority for component.
DAAC Manager	Integer	<10	Default = 0	(Sub) priority for DAAC Manager Users
Operator	Integer	<10	Default = 0	(Sub) priority for Operator Users
Researcher	Integer	<10	Default = 0	(Sub) priority for Researcher Users
Scientist	Integer	<10	Default = 0	(Sub) priority for Scientist Users
PGE Type	--	--	--	Strategy component name (Production Generation Executive)
Weight	Floating Pt number	<100.	Default = 00.00000	Weight for this component in final priority computation
Default	Integer	<10	Default = 5	Priority for component.
BTS#113#01	Integer	<10	Default =0	(Sub) priority for PGE BTS#113#01
Production Request Editor	--	--	--	Strategy component name (User that made the PR)
Weight	Floating Pt number	<100.	Default = 50.00000	Weight for this component in final priority computation – the priority is part of the PR.
Total Weight	Floating Pt number	< 100.	No Default	This field will be zero unless the user enters a weight in one of the 4 weight fields: PR Type Weight, User Type Weight, PGE Type Weight, or User Selected Weight. If that value than becomes > 100.0, hit the “Normalize” button to normalize.
Priority	Integer	< 10	Default = 1	Raise or lower this value by clicking on either the up or the down arrow beside that window.

Table 4.8.3-2. Production Strategies Field Descriptions (2 of 2)

Field Name on screen	Data Type	Size (# of characters)	Entry	Description
Type List	ASCII text	20	Default = blank	For an active type (ex., click on the PR Type button), the various valid types appear in this window. Inactive type and their priorities can be added using the "Add" button. Existing active types can be modified using the "Modify" button.
Late Start Delta Priority	Integer	<100	Default = 0	Priority for component.

4.8.3.2.1 Open Production Strategy Pop-up

The **Open Production Strategies** GUI shown in Figure 4.8.3-2 displays Production Strategies stored in the database. This pop-up is used to select the Production Strategy for display on the Production Strategies Main Screen.



Figure 4.8.3-2. Open Production Strategies Pop-up

The production strategies available in the PDPS database are displayed in the **Production Strategies** window. Selecting one of the displayed strategies will cause the selected strategy to be displayed in the File Selection window. The Open action is completed by clicking the “OK” button to accept the selection or the “Cancel” button to reject it.

4.8.3.2.2 Active Production Strategy GUI

The **Active Production Strategy** screen, shown in Figure 4.8.3-3, displays the default values from the PDPS database.

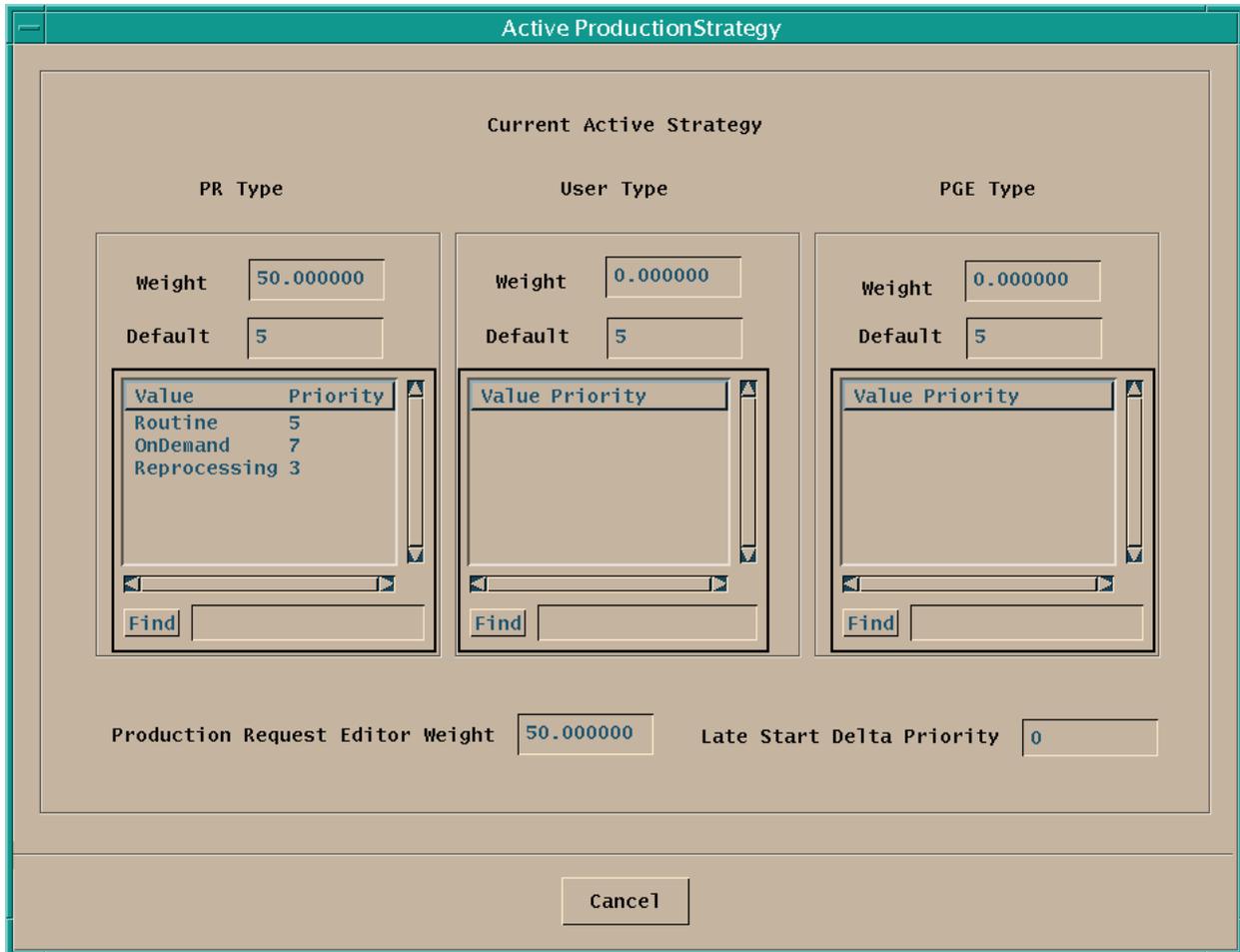


Figure 4.8.3-3. Active Production Strategy Screen

The fields displayed on the **Active Production Strategy** GUI are the same as those on the Production Strategies User Interface, Table 4.8.3-2.

4.8.3.3 Required Operating Environment

These GUIs may be hosted on the Planning Server and Queuing Server, Planning Workstation.

For information on the operating environment, tunable parameters and environment variables of Production Strategies User Interface, refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series. The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

Table 4.8.3-3 identifies the supporting products this tool depends upon in order to function properly.

Table 4.8.3-3. Support products for Production Strategies User Interface

Product
Sun Solaris 2.5
MOTIF 1.2.3
PDPS Database

4.8.3.3.1 Interfaces and Data Types

The Planning Subsystem has no interfaces external to ECS. It interfaces with the Data Processing Subsystem through the Planning and Data Processing System (PDPS) database to exchange planning information. The PDPS database is populated with information via the SSI&T components after the integration of a new or updated PGE. An element of the Planning system which does not require an operator interface, the Subscription Manager interfaces with the ECS Science Data Server Subsystem.

The Planning Subsystem is primarily intended for use by the operations staff of the EOSDIS DAACs. However, the SCFs have the capability to enter Production Requests into the Planning subsystem using the same interface as is described above.

4.8.3.4 Databases

The Planning Subsystem includes the PDPS database which contains information needed to plan the processing at a DAAC. It is also used for resource planning, as it contains information on the configured resources and their allocation. The database is also used by the Data Processing Subsystem, the other major element of the PDPS. The following subsections provide a summary description of the PDPS database table, and tools used with the database.

The SSI&T capability, which is a design component within the Data Processing Subsystem, includes capabilities to support the integration and test of the science software (PGEs) within the production processing environment. As a part of this activity, the SSI&T GUIs, described in Section 4.5 of this document, provide for the entry into the databases used by SSI&T of PGE-related information needed to plan for and run science software. At the conclusion of the SSI&T process, this database information is transferred to the operational databases from the databases established for SSI&T.

The complete database schema is listed in 311-CD-106-005, *Planning and Data Processing Subsystem Database Design and Database Schema Specifications for the ECS Project*.

4.8.3.5 Special Constraints

There are no special constraints that are associated with the Production Strategies User Interface tools.

4.8.3.6 Outputs

Outputs of the Production Strategies User Interface are provided in the following ways:

1. Production Strategies User Interface responses as described above.
2. Updates to the PDPS database (described in Section 4.8.3.3).

4.8.3.7 Event and Error Messages

The Production Strategies User Interface tool provides informational messages or warnings for minor errors. Significant events or errors are logged to the ECS Production Strategies Server ALOG file. The Error messages are listed in Appendix A.

4.8.3.8 Reports

The Production Strategies User Interface does not produce any reports.

4.9 Production Processing

This section describes the Production Processing tools used by DAAC operators:

1. AutoSys/AutoXpert
2. QA Monitoring Tool

This page intentionally left blank.

4.9.1 AutoSys/AutoXpert

This section describes how AutoSys and AutoXpert are used by DAAC Operations for Production Processing.

AutoSys is a job scheduling and management COTS product that provides mainframe batch scheduling functionality in a distributed UNIX environment. AutoSys has three primary components: AutoSys database, Event Processor, and Remote Agent. The AutoSys database is the repository for all system events, and for all job, monitor, and report definitions. The Event Processor is AutoSys' scheduling engine. It continuously queries the AutoSys database for events to be processed and performs necessary job, resource, and fault management actions. The Remote Agent is a transient process that is initiated by the Event Processor to run a program on the client machine. It sends messages directly to the Data Server to indicate changes in states for the job (e.g., STARTING, SUCCESS, FAILURE).

AutoSys is embedded in the ECS PDPS software specifically to manage the execution of jobs on science processing computers. Jobs that are necessary for the execution of DPRs are created and organized into "job boxes" using a custom API named "Job Management" in the ECS PDPS software.

Operator interaction is limited to "force starting" failed jobs from the Ops Console or from the JobScape GUI and monitoring the progress of jobs from JobScape. Entire job boxes should be canceled using the Planning Subsystem's Production Request Editor. Jobs should not be deleted using the AutoSys Job Definition GUI. This does not communicate with the PDPS database.

For uninterrupted processing, AutoSys supports automatic fault tolerance management via the High Availability option. In this configuration, multiple Event Servers (Primary and Secondary) and Event Processors (Primary and Shadow) can be setup for complete redundancy in case of anomalous situations such as hardware, software, or network problems.

AutoXpert, a companion tool for AutoSys, is an advanced graphical facility for monitoring jobs in AutoSys. Its components are HostScape, JobScape, and TimeScape. HostScape provides a view of the resources, showing active states of machines, Event Server(s), Event Processors(s), and jobs. JobScape presents a PERT-like view of job processing from a logical point of view. TimeScape presents a Gantt-like view of job processing from a temporal point of view. All three GUIs provide user configurable color codes via the X resources file for correlating unique colors to job and resource states.

AutoSys and AutoXpert are used to perform the operating functions listed in Table 4.9.1-1.

**Table 4.9.1-1. ECS Operator Functions Performed by AutoSys/AutoXpert
(1 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Start the Event Processor(s)	<code>eventor</code>	Starts the Event Processor (and optionally, the Shadow Event Processor).	To start the Event Processor(s). Normally, this would be done when the system is booted*
Startup GUI Console Panel	<code>/usr/ecs/<mode>/CUSTOM/bin/DPS/EcDpPrStartAutosys <AutoSys Instance Id></code>	Launches AutoSys GUI Control Panel.	To launch Ops Console, HostScope, JobScope, and TimeScope GUIs.
Access database	<code>xql</code>	Invokes direct Sybase database access utility to issue SQL commands.*	Used for troubleshooting purposes.*
Check overall system	<code>chk_auto_up</code>	Check the overall health of the system including the environment, configuration files, Event Server(s), and Event Processor(s).	To determine the overall health of the system for troubleshooting purposes.
Control job execution	JobScope	The GUI provides a display of job's progress.	To monitor jobs.
Determine active and completed jobs	Ops Console	View the list of successfully-completed jobs by selecting the Jobs Completed button.	To view the successfully completed jobs which have been removed from AutoSys.
Generate report	<code>autorep</code>	Generates detailed or summary format report of jobs	To report on the history of job processing.
Manage product license keys	<code>gatekeeper</code>	Manages product license keys.	To manage product license keys.*
Manage security	<code>autosecure</code>	Maintain Edit and Exec Superusers (Edit Superuser can edit any job definition, and Exec Superuser can execute any job and stop the Event Processor(s)).	To manage system security.*
Monitor and manage alarms	Alarm button from Ops Console, HostScope, JobScope, and TimeScope GUIs	View alarms, acknowledge them, and change the status.	To monitor and manage alarms.

**Table 4.9.1-1. ECS Operator Functions Performed by AutoSys/AutoXpert
(2 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Monitor and manage jobs	Ops Console button from GUI Control Panel	Compact "Control Center" for monitoring jobs and sending events to jobs: (Kill Job, Force Start Job, On Hold, Off Hold). Can also Launch Alarm Manager and view reports on all events sent to a job.	To monitor and manage jobs. See Controlling Job Execution in this section.
Monitor job history	TimeScope button from GUI Control Panel	To see an overall view of processing for jobs running in AutoSys. However, jobs that have finished successfully will have been removed, and jobs not ready to run will be waiting outside of AutoSys. See: Determine active and completed jobs.	To monitor current jobs running.
Monitor jobs by Job Box	JobScope button from GUI Control Panel	Viewing the progress of jobs in the job box. An Event GUI can be brought up by selecting a job and clicking the right mouse button.	To monitor status of a job.
Monitor resources	HostScope button from GUI Control Panel	Monitor machines, Event Server(s), Event Processor(s), and jobs.	To monitor resources and jobs. This GUI can also be used to check resource utilization to see if the processing load is being balanced across machines.
Perform synchronization between database and active/inactive jobs	chase	Verifies what AutoSys thinks is running, is actually running.	This command is run as part of the Event Processor(s) startup, and should not have to be run manually.*
Print AutoSys configuration information	autoflags	Prints AutoSys configuration information such as version and release number, database used, operating system, hostname, and hostid.	To retrieve necessary information to be supplied to the vendor for product license key generation.*
Remove events	archive_events	Removes events and alarms or job_runs information from the database.	This command is run automatically during the daily DB Maintenance cycle.*

**Table 4.9.1-1. ECS Operator Functions Performed by AutoSys/AutoXpert
(3 of 3)**

Operating Function	Command/Script or GUI	Description	When and Why to Use
Remove Remote Agent log files	clean_files	*Deletes old Remote Agent log files from client machines which have had jobs started on them.	This command is run automatically during the daily DB Maintenance cycle, and should not have to be run manually.*
Retrieve status of a job	autostatus	Retrieves the status of a job.	To monitor job status.*
Send event	sendevent from command line Send Event button from Ops Console	Sends an event to manage job execution, stop Event Processor(s), and cancel a scheduled event.	To stop the Event Processor(s), since everything else can be done from the GUIs.*
Verify machine	autoping	Verifies that both client and server machines are configured properly, and the client is functioning properly.	To verify a configured machine for troubleshooting purposes.
View job dependencies and conditions	job_depends	Generates a detailed report about the dependencies and conditions of a job.	To view job dependencies and conditions.*
View job queue	Ops Console - Jobs Waiting Button	Used to view jobs in CQ_RELEASED, CQ_HOLD and JIL_FAILUR (either the job is already in AutoSys, or AutoSys environment variables are invalid) states by selecting the Jobs Waiting button.	Verify that jobs have been queued or determine why a queued job was not released.
View successfully-completed jobs	Ops Console - Jobs Completed Button	Used to view jobs that have successfully completed and have been removed by Job Management from AutoSys.	Used to view list of jobs that have finished but are not in AutoSys.
View processing log files	autolog	Viewing the Event Processor log file is an alternative to monitoring all jobs and events using the GUIs.	To monitor jobs, events, and alarms for troubleshooting purposes.*
* Not generally used in normal operations			

4.9.1.1 Quick Start Using Autosys/AutoXpert

To invoke AutoSys/AutoXpert from the command line prompt, enter:

>EcDpPrAutosysStart <mode> <AutoSys Instance Id>

where:

<mode> is the ECS mode which this AutoSys instance is associated, e.g., OPS, TS1.

<**AutoSys Instance Id**> is the instance name of this AutoSys.

This script sets the appropriate AutoSys environment variables and issues the **autosvc** command to start the GUI Control Panel. AutoXpert can not be invoked by a command line interface. Its functions are invoked by buttons on the AutoSys Main Screen.

The documentation of AutoSys used as a basis and referenced in this section is for version 3.3.7, contained in ECS Release 4.

The documentation of AutoXpert used as a basis and referenced in this section is for version 3.4, contained in ECS Release 4. The AutoXpert documentation may be found at the Internet URL: **<http://support.platinum.com/tech/docs.html>**.

Refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series , for a listing of the EcDpPrAutosysStart script.

AutoSys provides various scripts and executables for defining objects, checking system status, accessing the database, monitoring and reporting job status, defining custom calendars, recording sounds, generating license keys, and controlling system security (see *AutoSys User Manual*, Chapter 11 “Using AutoSys Commands”).

4.9.1.2 AutoSys Main Screen

The AutoSys GUI Control Panel provides buttons for launching (bringing up) Ops Console, Job Definition, Calendars (not used), Monitor/Browser (not used), HostScape, JobScape, and TimeScape GUIs. Figure 4.9.1-1 shows the AutoSys GUI Control Panel.

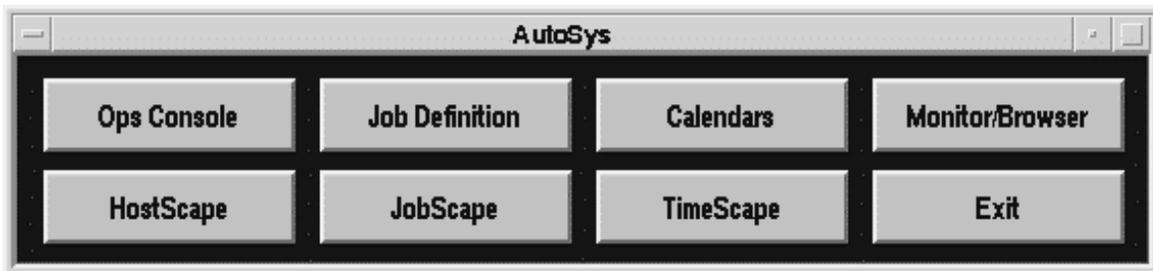


Figure 4.9.1-1. AutoSys GUI Control Panel

For more information about the AutoSys GUI Control Panel please see the *AutoSys Users Manual*, Chapter 5 Defining AutoSys Jobs using the GUI.

4.9.1.2.1 AutoSys Ops Console

The AutoSys Ops Console GUI provides an interface to monitor jobs and alarms in real-time. The GUI provides a Job Selection Dialog for filtering jobs based on various parameters such as type, name, state, and machine. Detailed information about the selected job including starting

conditions, dependent jobs, and reports can be viewed. In addition, there are buttons for launching Alarm Manager, Job Definition, and Send Event GUIs.

For more information about the AutoSys Ops Console please see the *AutoSys Users Manual*, Chapter 9 The Operator Console.

4.9.1.2.1.1 ECS-Added Functions to the AutoSys Ops Console

ECS has added the following three buttons to the AutoSys Ops Console:

- Jobs Completed
- Jobs Waiting
- Job Management Client Tool

4.9.1.2.1.1.1 Jobs Completed Display

When this button is selected a display similar to the following appears on the screen:

```
#####  
##### JOBS COMPLETED #####  
#####
```

DPR ID	COMPLETION STATE	PRIORITY
MODPGE01#s02280100DEV04	SUCCESS	250
MODPGE02#s28010500DEV04	SUCCESS	250
MODPGE02#s28011000DEV04	SUCCESS	250
MODPGE02#s28011500DEV04	SUCCESS	250
MODPGE02#s28012000DEV04	SUCCESS	250

These jobs have finished successfully and have been removed from AutoSys so that the database is optimally maintained to increase job throughput. Note that only jobs that have successfully completed are automatically removed from AutoSys under program control. The number of jobs that can exist in AutoSys is controlled by the DPS Job Management configuration parameter, `DpPrAutoSysMaxJobs`. The oldest successfully completed jobs are removed once this limit has been reached.

4.9.1.2.1.1.2 Jobs Waiting Display

Select this button to display a list of jobs that have been scheduled, but which are waiting for data. When all of the data is available for a job, jobs will be released from this holding queue into AutoSys where they will immediately start to execute.

4.9.1.2.1.1.3 Job Management Client Tool

The Job Management Client Program can be used to manually override the normal PDPS operations of creating jobs from information in the PDPS database and removing them after they have been successfully completed. *Except for creating and deleting ground event jobs (which must be invoked via this tool), this capability is almost always used only during system testing. You would only need to use this tool if a problem was identified and you were waiting for it to be resolved through normal channels.* You must first select a job box (not a job contained within a job box) to identify a DPR to work with. The following menu is displayed after selecting the **JM Client** button from the AutoSys Ops Console:

```
*** Current DPRId:MODPGE08#s28015500DEV04 Current Mode:DEV04 ***
```

- 0) Exit
- 1) Create DPR Job
- 2) Release DPR Job
- 3) Cancel DPR Job
- 4) Change DPR Id
- 5) View Job Management DPR Queue
- 6) Create Ground Event Job
- 7) Cancel Ground Event Job

enter an option:

Once into the program, you can change DPR IDs by selecting option 4). This DPR ID stays in effect until another DPR ID is entered. Options 1) and 2), taken together, create and put a job into AutoSys, or on the queue, if AutoSys is full (the job will be in a CQ_RELEASE state). Option 3 can be used to cleanly (as opposed to manually deleting the job from AutoSys) remove a job from AutoSys.

4.9.1.2.2 AutoSys Job Definition

The AutoSys Job Definition GUI provides an interface to add new jobs, delete existing jobs, and modify attributes for existing jobs. Both Date/Time Options and Adv Features Dialogs can be used to set/modify all job attributes.

For more information about the AutoSys Job Definition please see the *AutoSys Users Manual*, Chapter 5 Defining AutoSys Jobs using the GUI.

Note: This capability must never be used without first checking with someone who knows something about the system internals. The ECS Job Management software handles the creating, running and removing of successfully-completed jobs. Manually performing these functions

bypasses Job Management housekeeping and may cause the processing system to get out of sync, possibly with severe consequences.

4.9.1.2.3 AutoSys Calendars

Not used by ECS.

4.9.1.2.4 AutoSys Monitor/Browser

Not used by ECS.

4.9.1.2.5 AutoSys Alarm Manager

The AutoSys Alarm Manager GUI provides an interface to monitor and manage alarms. Alarms can be filtered based on type, state, and time; a response can be registered for an alarm; and the state of the alarm can be changed. The Alarm Manager is invoked by clicking on the “Alarm” buttons on either the JobScape or Operator’s Console GUIs.

For more information about the AutoSys Alarm Manager please see the *AutoSys Users Manual*, Chapter 9 The Operator Console.

4.9.1.2.6 AutoXpert HostScape

The AutoXpert HostScape GUI provides an interface to monitor the states of resources, Event Server(s), Event Processor(s), and jobs. This GUI also provides buttons for launching Alarm and Job Console GUIs. For more information about HostScape see *AutoSys/Xpert User Guide for UNIX Version 3.4, Product Overview*.

4.9.1.2.7 AutoXpert JobScape

The AutoXpert JobScape GUI provides an interface to monitor states of jobs. This GUI presents a Pert-like (network) view of job processing from a job dependency point of view. Alarm and Job Console GUIs can be launched from this GUI. For more information about JobScape see *AutoSys/Xpert User Guide for UNIX Version 3.4, Product Overview*.

4.9.1.2.8 AutoXpert TimeScape

The AutoXpert TimeScape GUI provides an interface to monitor states of jobs. This GUI presents a Gantt-like (chart) view of job processing from a time-related point of view. This GUI also provides buttons for launching Alarm and Job Console GUIs. For more information about TimeScape see *AutoSys/Xpert User Guide for UNIX Version 3.4, Product Overview*.

4.9.1.3 Required Operating Environment

For all COTS packages, appropriate information on operating environments, tunable parameters, environment variables, and a list of vendor documentation can be found in a CM controlled

document for each product. To find the documentation for AUTOSYS, refer to the ECS Baseline Information System web page,

URL <http://cmdm.east.hitc.com/>.

No tuning or configuration of the product is required, beyond that required by a normal installation.

4.9.1.3.1 Interfaces and Data Types

AutoSys exchanges data of various types through interfaces within ECS. Table 4.9.1-2 lists AutoSys system interfaces for Version 2.0.

Table 4.9.1-2. Interfaces Between AutoSys and other ECS PDPS Components

Interface	Type of Primary Interface Protocols	Type of Backup Interface Protocols	Comments
ECS PDPS Subscription Manager	ECS PDPS Job Management/API	None	Interface between Planning and Data Processing subsystems for releasing Data Processing Request jobs in AutoSys
ECS PDPS Planning Workbench	ECS PDPS Job Management/API	None	Interface between Planning and Data Processing internal software subsystems for adding, deleting, and canceling Data Processing Request and Ground Event jobs in AutoSys

The information in the above table is informational only. It is not needed for normal operations.

4.9.1.4 Databases

All AutoSys information is stored in one of the three Sybase database types. The three types of databases are: Event Server, Monitor Server, and Alarm Server. The Event Server contains all the information about a particular instance of AutoSys such as job definitions, events, monitor and browser definitions, calendar information, and machine definitions. Monitor and Alarm Servers, not used within the ECS, are specialized databases for monitoring events and alarms.

See *AutoSys User Manual*, Chapter 13 “AutoSys Databases” for database schema definitions.

4.9.1.5 Special Constraints

None.

4.9.1.6 Outputs

AutoSys Event Processor(s), Event Server(s), and Remote Agent maintain log files. Table 4.9.1-3 lists AutoSys output log files.

Table 4.9.1-3. Outputs

Output	Description and Format
\$AUTOUSER/out/event_demon.\$AUTOSERV	Event Processor log file
\$SYBASE/install/errorlog	Event Server error log file
<i>AutoRemoteDir/auto_rem.joid.run_num.ntry</i>	Remote Agent log file

4.9.1.7 Event and Error Messages

AutoSys issues both status and error messages to the Event Processor log file (\$AUTOUSER/out/event_demon.\$AUTOSERV).

4.9.1.8 Reports

AutoSys autorep utility provides a method of generating reports containing information about jobs, machines, and global variables currently defined in the database. Table 4.9.1-4 lists AutoSys reports that can be generated. See the *AutoSys User Manual* for further information.

Table 4.9.1-4. Reports

Report Type	Report Description	When and Why Used
job	The report provides information about jobs and their processing status Summary, detail, and query formats can be generated	It should be used primarily for generating production history reports, but also can be used for real-time monitoring of jobs
machine	The report provides status of machines and attributes such as maximum load, current load, and factor Summary, detail, and query formats can be generated	It should be used for real-time monitoring of machines

4.9.1.8.1 Sample Report

Figure 4.9.1-2 shows an autorep job report obtained by using the command:

```
autorep - d -J % > report
```

Job Name Pri/Xit	Last Start	Last End	Status	Run	
BTS#syn1#020704981301	12/11 17:19	12/11 21:08	SUCCESS	9837/1	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
RUNNING	12/11 17:19:01	1	Processed	12/11 17:19:19	
SUCCESS	12/11 21:08:54	1	Processed	12/11 21:08:59	
BTS#syn1#020704981301A	12/11 17:56	12/11 18:00	SUCCESS	9837/2	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
STARTING	12/11 17:19:15	1	Processed	12/11 17:19:19	pdps1
RUNNING	12/11 17:19:19	1	Processed	12/11 17:19:24	pdps1
FAILURE	12/11 17:19:48	1	Processed	12/11 17:19:50	
[*** ALARM ***]					
JOBFAILURE	12/11 17:19:50	1	Processed	12/11 17:19:56	pdps1
[FORCE_STARTJOB]	12/11 17:56:19	0	Processed	12/11 17:56:27	
STARTING	12/11 17:56:24	2	Processed	12/11 17:56:27	pdps1
RUNNING	12/11 17:56:28	2	Processed	12/11 17:56:32	pdps1
SUCCESS	12/11 18:00:51	2	Processed	12/11 18:00:58	
BTS#syn1#020704981301S	12/11 18:00	12/11 18:11	SUCCESS	9837/1	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
STARTING	12/11 18:00:55	1	Processed	12/11 18:00:58	pdps1
RUNNING	12/11 18:00:59	1	Processed	12/11 18:01:10	pdps1
SUCCESS	12/11 18:11:25	1	Processed	12/11 18:11:32	
BTS#syn1#020704981301P	12/11 19:02	12/11 19:02	SUCCESS	9837/2	
Status/[Event]	Time	Ntry	EventState	ProcessTime	Machine
STARTING	12/11 18:11:29	1	Processed	12/11 18:11:33	pdps1
RUNNING	12/11 18:11:33	1	Processed	12/11 18:11:38	pdps1
SUCCESS	12/11 18:13:20	1	Processed	12/11 18:13:27	
[FORCE_STARTJOB]	12/11 19:01:56	0	Processed	12/11 19:02:02	
STARTING	12/11 19:02:02	2	Processed	12/11 19:02:07	pdps1
RUNNING	12/11 19:02:04	2	Processed	12/11 19:02:07	pdps1
SUCCESS	12/11 19:02:22	2	Processed	12/11 19:02:28	

Figure 4.9.1-2. AutoSys autorep job report

Figure 4.9.1-3 shows an autorep job report obtained by using the command:

```
autorep -d -m ALL >report
```

Machine Name	Max Load	Current Load	Factor	O/S
chimera	---	---	1.00	Unix
deepsea	---	---	1.00	Unix
guinness	---	---	1.00	Unix
lasher	---	---	1.00	Unix
monet	---	---	1.00	Unix
nessie	---	---	1.00	Unix
pdps1	---	---	1.00	Unix
string1.pdps1	1	0	----	Unix
pdps2	---	---	1.00	Unix
string1.pdps2	2	0	----	Unix
relbsgi	---	---	1.00	Unix
sdps2	---	---	1.00	Unix
seagull	---	---	1.00	Unix
Current Jobs:				
Job Name	Machine	Status	Load	
Priority				
-----		-----	-----	-----
ETS#syn1#020704971301		RUNNING	1	250

Figure 4.9.1-3. AutoSys autorep job report - all

4.9.2 ECS Quality Assurance (QA) Monitor

ECS QA Monitor processing capabilities enable DAAC operators to perform duties associated with DAAC QA activities. The ECS Quality Assurance (QA) Monitor GUI is the user-interface for entering data requests and displaying data, status, and error messages. The QA Monitor does not produce data products, but communicates with the science data server to retrieve data that have been previously archived.

The QA Monitor GUI is used to perform the operator functions listed in Table 4.9.2-1.

Table 4.9.2-1. Common ECS Operator Functions Performed with QA Monitor

Operating Function	Description	When and Why to Use
Query data granules	Initiates a request to search the science archive for data granules within a date ¹ interval	When there is a need to know all archived data granules with the same data type and inserted in the archive at a certain time (date interval)
Retrieve data granules	Initiates a request to get data granules from the science archive	When data granule(s) needs to be transferred from archive to local disk for visualization
Visualize data (HDF files)	Display Visualize screen	When graphical images of data granules need to be viewed to assess quality
Update metadata	Initiates a request to archive QA information about data granules	When QA information about data granules needs to be updated in the archive based on DAAC QA activities encompassing use of the Visualize Data function.

4.9.2.1 Quick Start Using QA Monitor

The QA Monitor is used to search for granules in the science archive, to retrieve granules and/or processing history files from the science archive and to update metadata of granules retrieved from the science archive. The QA Monitor GUI's main screen has three windows: a window that displays the ESDTs of granules in the science archive, a data granule display window to select data to browse, and a status window to indicate progress or display error messages. Several dialog screens may be activated when functions are performed. Some windows are display-only for informational text, while others require user-interaction.

4.9.2.1.1 Invoking QA Monitor Tool From the Command Line Interface

To execute QA Monitor from the command line prompt use the startup script provided during installation and located in utilities subdirectory of the operational directory structure:

```
EcDpPrQaMonitorGUIStart <mode>
```

¹ Data acquisition beginning and ending date.

<mode> is the ECS Mode for the execution, e.g., OPS, TS1.

Refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series , for a listing of EcDpQaMonitorGUIStart.

4.9.2.2 QA Monitor Main Screen

The QA Monitor Tool Main Screen shown in Figure 4.9.2-1 provides access to all QA Monitor primary functions. It consists of the QRU (Query/Retrieval/Update) data Tab and the Visualize data Tab. The QRU data Tab is activated when the QA Monitor Main Screen is initialized. A group of related objects is displayed by default to help the operator get started. The Data Types, Data Granules, Date Granule Insert (acquisition date of granules) windows, and the Query pushbutton are displayed, plus four desensitized (grayed-out) pushbuttons. These objects and the QRU data object which controls display of this screen are needed to perform a query. Other available data types are displayed in the Data Types window so the operator can choose which data type to query on. This enforces the sequence of events: query, retrieve and update. The grayed-out pushbuttons are sensitized after a Query, which permits selection of the indicated functions. The Visualize data Tab displays a list of science granules available for browsing.

From this screen, the following functions can be performed:

- Request the Science Data Server to search for specific types of Data Granules (Section 4.9.2.2.1)
- Request the Science Data Server to transfer Data Granules to the operator’s computer
- Request the Science Data Server to transfer Production History to the operator’s computer
- Update Data Granules’ Metadata (Section 4.9.2.2.2)
- View graphical images of Data Granules (Section 4.9.2.2.3)
- Print Display Lists of Data Granules and Data Types (Section 4.9.2.2.4)

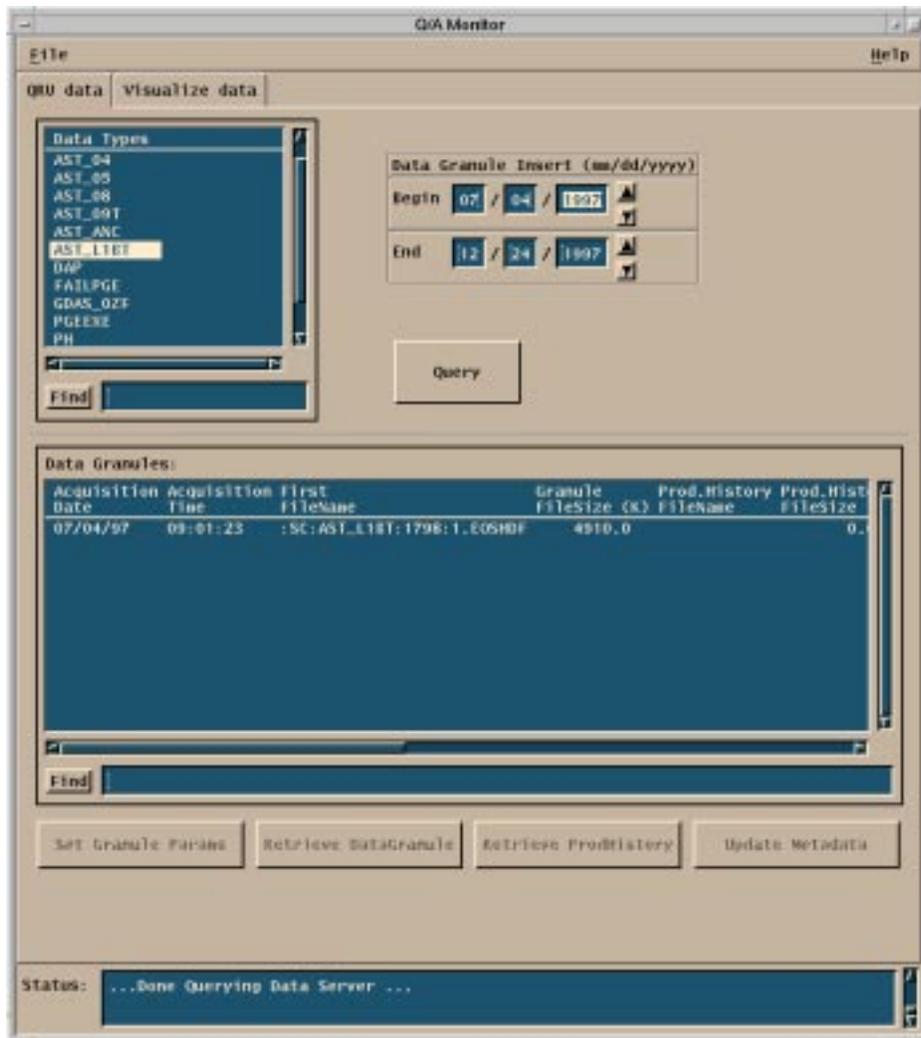


Figure 4.9.2-1. QA Monitor Tool Main Screen

The operator can select the following menu items from the QA Monitor Main Screen

- **File** This menu contains the following items:
 - Print** allows the operator to print the Data Types list and the Data Granules list (Section 4.9.2.2.4)
 - Exit** allows the operator to exit the QA Monitor
- Help** The "Help" menu offers the following help choices: (Note: Presently not functional)
 - On Help** displays help about using help
 - On Context** displays help about individual screen elements. Context sensitive help pops up a scrolled window that displays text about a specific user interface component. There are two ways to get context-sensitive help:

The first way to get context sensitive help is to move focus to an active user interface component (e.g., text entry field, scroll list) and press the Help key. Help will be displayed about that element. If the keyboard does not have a key labeled Help, the F1 key usually functions as the help key.

The second way is to select "On Context" from the Help menu. The cursor will change to a question mark. Clicking on any user interface component will display help about that element.

On Window displays high-level help about the window (see "Tutorial" below)

On Keys displays help about using the keyboard and general help about standard user interface components that support complex behaviors. Topics include:

1. how to get context sensitive help
2. how to use keyboard mnemonics and accelerators to access menu functions without using the mouse
3. how to navigate between active fields on windows using the keyboard

Index is currently not supported.

Tutorial displays an extended discussion of the application and its role within the ECS system.

On Version identifies what version of ECS software is currently being used.

On Mode identifies the operating mode of the ECS software currently being used.

Additionally, the following functions may be selected on the QRU data Tab:

- **Query:** refer to Section 4.9.2.2.1
- **Set Granule Params:** refer to Section 4.9.2.2.2
- **Retrieve Data Granule:** refer to Section 4.9.2.2.1
- **Retrieve ProdHistory:** refer to Section 4.9.2.2.1
- **Update Metadata:** refer to Section 4.9.2.2.2

The following functions may be selected on the Visualize data Tab (refer to Section 4.9.2.2.2):

- **Visualize:** refer to Section 4.9.2.2.3
- **Help:** refer to Section 4.9.2.2.3

4.9.2.2.1 Query and Retrieve Data

The QA Monitor allows the operator to search the data archive for data granules by data type. The available data types are displayed when the main screen comes up and allows the operator to

choose which data type to query on. The operator may accept the default dates or enter new dates to further constrain the search. Clicking on the Query pushbutton initiates the search request. If a search is successful a list of data granules are displayed. Then the operator can have them transferred from the data archive to his computer. Where the data will be located on the operator's computer is a configurable item. To request transfer of a data granule, highlight the row with information about the granule desired by clicking on it then click on the Retrieve Data pushbutton. A message will appear in the Status window indicating the processing status. To get the production history file associated with the granule just click on the Retrieve ProdHist pushbutton.

For a description of the "Query and Retrieve Data" fields, see Table 4.9.2-2 below.

Table 4.9.2-2. QA Monitor Field Descriptions

Field Name	Data Type	Size	Entry	Description
Data Granule Insert Begin End	date min max	8 8	initial default value(1/2/1901-6/1/2036) can be changed by operator	The date interval window that data granules are inserted into the data server
Data Types list	single selection	N/A	operator selects a data type from the list displayed at startup	The list of all available data types at a specific DAAC
Data Granules list	multiple selection	N/A	operator clicks data granule row(s) then clicks retrieve pushbutton	The list of all data granules in the date interval above for a particular selected data type are available for retrieval
Status	text	N/A	display status messages only	Display status messages

In addition, the following pushbuttons are provided:

- **Query** populates the list of data granules on the bottom half of the screen for a particular selected data type within a date interval.
- **Find** (below the data types and data granules list) allows the operator to perform a keyword search for information stored in those 2 lists.
- **Retrieve DataGranule** allows the operator to retrieve data granule(s) from the DAAC's data archive to local disk.
- **Retrieve ProdHistory** allows the operator to retrieve a production history tar file from the DAAC's data archive to local disk.
- **Update Metadata** pops up a Granule Parameters screen (see Section 4.9.2.2.2 below).

4.9.2.2.2 Update Metadata

Clicking on one data granule displayed in the Data Granules Field and then the Update Metadata pushbutton brings up the Granule Parameters screen shown in Figure 4.9.2-2. The list of parameter names are attributes describing the selected data granule. The settings for the parameters are determined by the DAAC, SCF or PGE. Those settings are displayed along with explanations, if any. After DAAC QA of a data granule the DAAC operator may update the operational QA flag providing an explanation. To update a parameter, click the parameter name then click OK. The Update Metadata screen will be displayed as shown in Figure 4.9.2-3.

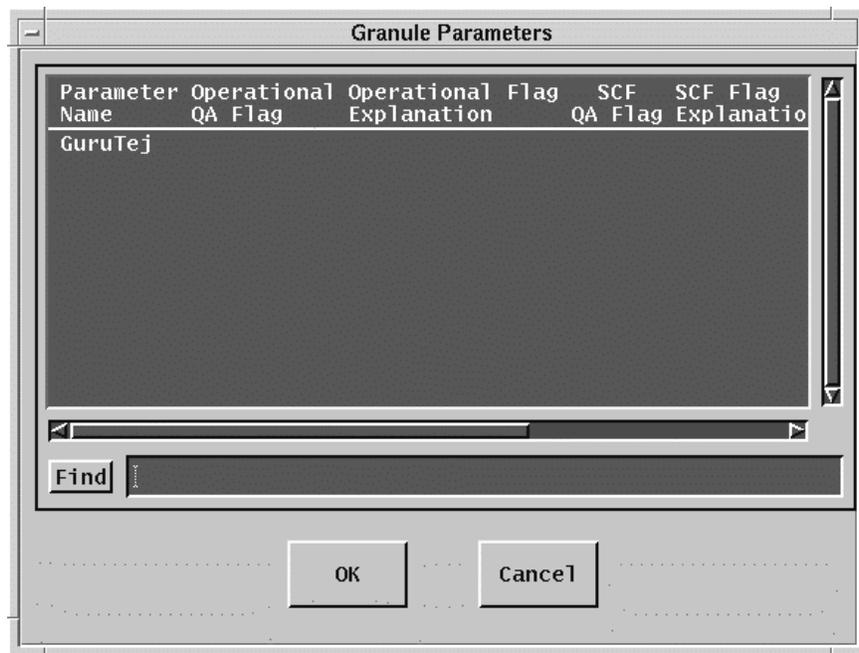


Figure 4.9.2-2. Granule Parameters Dialog

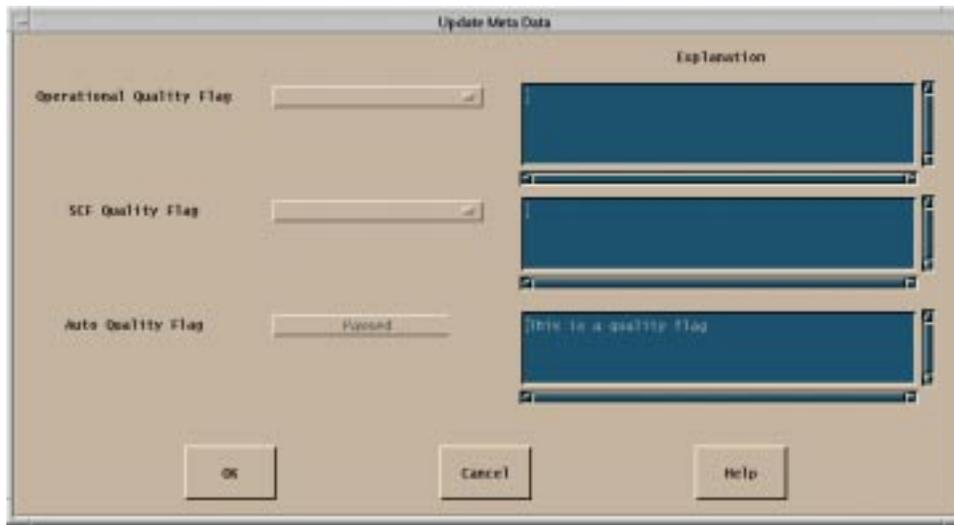


Figure 4.9.2-3. Update Metadata Dialog

For a description of the “Update Metadata” dialog fields, see Table 4.9.2-3 below.

Table 4.9.2-3. Update Metadata Field Descriptions (1 of 2)

Field Name	Data Type	Size	Entry	Description
Operational Quality Flag, SCF Quality Flag	character	N/A	operator selects value	DAAC and SCF quality status setting of a data granule parameter. The valid values are: <ul style="list-style-type: none"> • passed • failed • being investigated • not being investigated
Operational Quality Explanation, SCF Quality Explanation	character	Less than 150 chars	Operator inputs text	Text describing quality status
Auto Quality Flag	character	N/A	Operator input not allowed	Quality status of a data granule parameter set during data processing. The valid values are: <ul style="list-style-type: none"> • passed • failed • being investigated • not being investigated

Table 4.9.2-3. Update Metadata Field Descriptions (2 of 2)

Field Name	Data Type	Size	Entry	Description
Automatic Quality Explanation	character	N/A	Operator input not allowed	Quality status of a data granule parameter set during data processing.

In addition, the following pushbuttons are provided:

- **OK** sends an update request to the data server to update data granule parameter values.
- **Cancel** will cancel the update request.
- **Help** displays the help for Update Metadata.

4.9.2.2.3 Visualize Data

Clicking on Visualize data from the QA Monitor GUI main screen brings up the **Visualize Data Tab** shown in Figure 4.9.2-4. The operator clicks a row with the name of a science file to browse. When the Visualize pushbutton is activated EOSView is invoked. The EOSView GUI requires operator input to produce a graphical image of the science data file (see Section 4.12.7 “EOSView”). Note that the EOSView GUI can only read data products that are in HDF format.

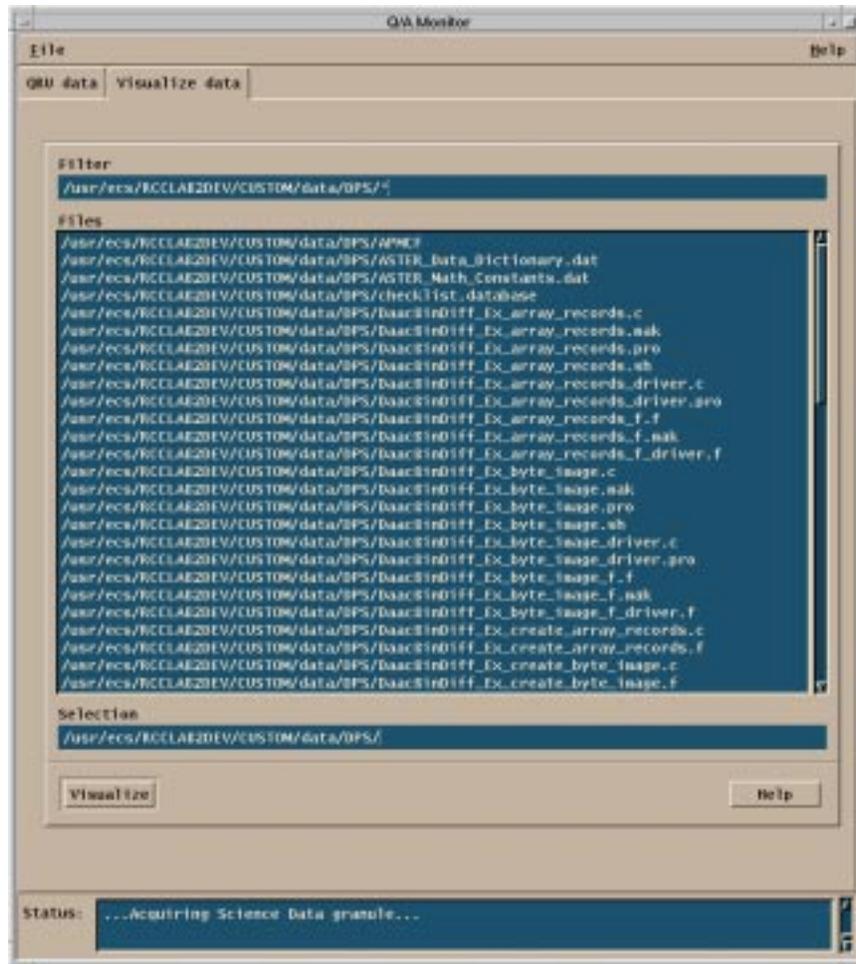


Figure 4.9.2-4. Visualize Data Tab Stack

For a description of the “Visualize data” fields, see Table 4.9.2-4 below.

Table 4.9.2-4. Visualize Data Field Descriptions

Field Name	Data Type	Size	Entry	Description
Filter	character	N/A	uneditable	Indicates path and filenames to be displayed as established in the configuration file
Files	character	N/A	click on a row	Highlights row information
Selection	character	N/A	uneditable	Full pathname of the file selected for viewing will be displayed

In addition, the following pushbuttons are provided:

- **Visualize** invokes EOSView for visualizing data products.
- **Help** displays help for the file selection box.

4.9.2.2.4 Print Options

Clicking on “File” pull-down menu brings up the **Print Options** dialog shown in Figure 4.9.2-5. This dialog allows the operator to select which list, data types list or data granules list, or both, to be sent to the printer.

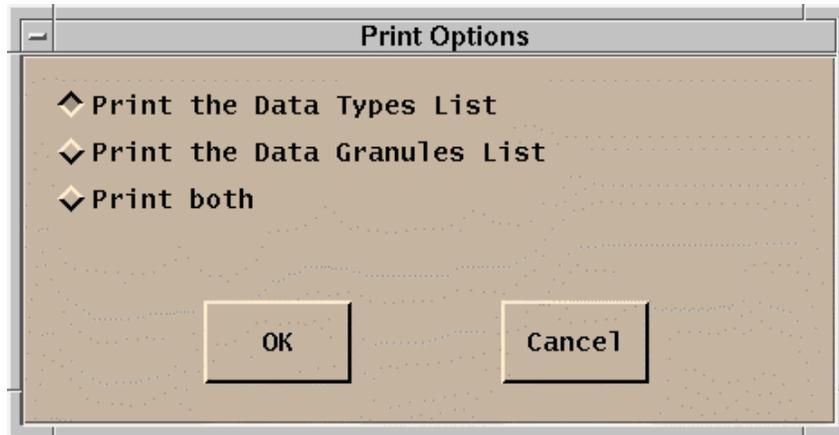


Figure 4.9.2-5. Print Options Dialog

For a description of the “Print Options” fields, see Table 4.9.2-5 below.

Table 4.9.2-5. QA Monitor Tool Field Descriptions

Field Name	Data Type	Size	Entry	Description
Print data types list	N/A	N/A	select 1 at a time	This toggle button is selected if the operator wants the data types list to be sent to the printer.
Print data granules list	N/A	N/A	select 1 at a time	This toggle button is selected if the operator wants the data granules list to be sent to the printer
Print both	N/A	N/A	select 1 at a time	This toggle button is selected if the operator wants both lists to be sent to the printer

In addition, the following pushbuttons are provided:

- **OK** sends a print request to the printer for option selected.
- **Cancel** will cancel the print request.

4.9.2.3 Required Operating Environment

For information on the operating environment, tunable parameters and environment variables of The QA Monitor Tool refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series . The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC.

4.9.2.3.1 Interfaces and Data Types

The QA Monitor Tool does not have any external interfaces to ECS. However, there is an internal interface to the Science Data Server to query and retrieve data granules, and update metadata. There is also an internal interface to the Advertising server to get all data.

4.9.2.4 Databases

None

4.9.2.5 Special Constraints

None.

4.9.2.6 Outputs

Other than the output displayed to the operator the QA Monitor is responsible for file (science data files in the HDF format) transfers from the science archive to the local disk of the QA Monitor host computer. It also sends data to the science archive to be stored.

4.9.2.7 Event and Error Messages

The QA Monitor issues status messages to the Status Messages text area at the bottom of the QA Monitor GUI and error messages on pop-up error dialogs. Both event and error messages are listed in Appendix A.

4.9.2.8 Reports

None.

This page intentionally left blank.